Insights into cosmological structure formation with machine learning

Luisa Lucie-Smith

Submitted for the degree of Doctor of Philosophy

Department of Physics and Astronomy

University College London

Supervisors:

Prof. Hiranya V. Peiris

Prof. Andrew Pontzen

Examiners:

Prof. Benjamin Wandelt

Prof. Ofer Lahav

31st of October, 2019

I, *Luisa Lucie-Smith*, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Abstract

Our modern understanding of cosmological structure formation posits that small matter density fluctuations present in the early Universe, as traced by the cosmic microwave background, grow via gravitational instability to form extended haloes of dark matter. A theoretical understanding of the structure, evolution and formation of dark matter haloes is an essential step towards unravelling the intricate connection between halo and galaxy formation, needed to test our cosmological model against data from upcoming galaxy surveys.

Physical understanding of the process of dark matter halo formation is made difficult by the highly non-linear nature of the haloes' evolution. I describe a new approach to gain physical insight into cosmological structure formation based on machine learning. This approach combines the ability of machine learning algorithms to learn non-linear relationships, with techniques that enable us to physically interpret the learnt mapping. I describe applications of the method, with the aim of investigating which aspects of the early universe density field impact the later formation of dark matter haloes. First I present a case where the process of halo formation is turned into a binary classification problem; the algorithm predicts whether or not dark matter 'particles' in the initial conditions of a simulation will collapse into haloes of a given mass range. Second, I present its generalization to regression, where the algorithm infers the final mass of the halo to which each particle will later belong. I show that the initial tidal shear does not play a significant role compared to the initial density field in establishing final halo masses. Finally, I demonstrate that extending the framework to deep learning algorithms such as convolutional neural networks allows us to explore connections between the early universe and late time haloes beyond those studied by existing analytic approximations of halo collapse.

Impact Statement

This thesis presents a new approach based on machine learning, aimed at deepening our understanding of the formation of dark matter haloes in the Universe. Our goal is to understand what information is learnt by the machine learning algorithm about the underlying connection between the early universe and the late-time dark matter haloes in cosmological simulations; this differs from common approaches where machine learning is utilized as a black-box tool to obtain fast and automated mappings. Our method led to a re-interpretation of the existing understanding of halo formation over the last decades, in particular in relation to the role of the tidal shear tensor in establishing the final mass of dark matter haloes (Chapters 3 & 4). This work achieved academic impact through two scientific publications, cited by independent researchers around the world, and over ten professional presentations, including invited talks, at international conferences for broad and specialized audiences from the cosmology and machine learning communities. Thanks to the broad applicability of our method, there has been an upsurge of interest in the community to apply our method to other aspects of dark matter haloes, leading to new international collaborations for the PhD candidate.

The advances in machine learning presented in this thesis can be directly applied to industrial and commercial applications of artificial intelligence (AI). One of the key problems faced in the AI community is the issue of interpretability; without a deeper understanding of how deep learning algorithms make their predictions, we cannot trust AI tools in scientific, industrial and commercial applications. Our methods are specifically designed to turn black-box algorithms into interpretable ones, allowing one to better understand the complex systems these algorithms describe. This is particularly relevant for convolutional neural networks (Chapter 5), which are used extensively in industry. In addition to being interpretable, our deep learning architecture allows for broader

applicability to three-dimensional data sets than conventional architectures applied to 2D images.

Finally, this thesis points toward a new branch of machine learning research known as knowledge extraction (Chapter 6), where deep learning algorithms are constructed in a manner that allows for the discovery of fundamental properties of the underlying data sets. This work has initiated an international collaboration of experts in the fields of brain sciences, computational psychiatry, crime sciences and physics, as well as Google DeepMind. We expect this collaboration to expedite progress in the ability of humans to extract knowledge from machine learning algorithms.

Acknowledgements

First, I owe my deepest gratitude to my amazing supervisors, Hiranya Peiris and Andrew Pontzen, for investing an infinite amount of time in making me a better researcher. Thank you for always being there for me, for being extremely attentive mentors, for the endless support and understanding at difficult times and for sharing with me your scientific expertise and insights. Working with you and being your student has been a privilege. Thank you to all the other brilliant cosmologists I was lucky enough to collaborate with – Nina Roth, Michelle Lochner, Brian Nord, Risa Wechsler and Susmita Adhikari – for the all the help in the fun projects carried out together. I am extremely grateful to Edd Edmondson and John Deacon for solving all my IT problems in no time. Special thanks also go to Corentin Cadou, Martin Rey and Sam Witte for proof reading parts of this thesis.

Thank you to all the friends I have made at UCL for the laughs, pub trips and occasional trips to Borderline. Special thoughts go to Felix Priestley for listening to extensive monologues starting with 'about me now', Arthur Loureiro for keeping me up to date on Brazilian politics, Arianna Sorba for always having an answer to my questions, Niall Jeffrey for that magnum I still owe him, Martin Rey for our morning coffees and paper discussions, Keir Rogers for the wisest advices on surviving a PhD, Michelle Lochner for buying all the rounds at the pub, Andreu Font-Ribera for rigorously drinking only half pints at the pub, Roger Wesson for still not accepting that I won, Jonathan Braden for laughing at the words baryons and stars, Davide Gualdi for excellent coffees in his office, Will Hartley for always reminding us to never work on photo-z and Sam Witte, for introducing me to my newest big loves, Jim Croce and pad thai. Thank you to all the very good friends I made during my time at Fermilab, especially for the numerous pitchers of Aperol spritz drank at the village pub. Special thanks go to Antonella Palmese for making me feel at home and Federico Speranza for our shared passion for country music and for grating parmigiano on my pasta. Thanks to all the Italian

songwriters that kept me company during long hours of work, Francesco De Gregori, Lucio Battisti, Brunori Sas and Tiromancino, and to Pyotr Ilyich Tchaikovsky, for the most productive hours of thesis writing.

A big thank you and all my love goes to my family. Thanks to my dad, for teaching me special relativity when I was way too young to understand it, my brother, for always being there for all of us, and my sister for the “sclero times” together. Thank you to Martina for being my best friend. My deepest love goes to my mum, Antonella, for being the strongest and most inspiring woman I know on this planet and for all the love and support she has given me. Thank you for your efforts in trying to learn about my research, although I know you still think I work on black holes. I dedicate this thesis to you, cocca.

Contents

List of Figures

List of Tables

1 Introduction

1.1 The Universe

The field of modern cosmology, studying the origin and evolution of our Universe as a whole, began with the discovery of the expansion of the Universe in 1929 (Hubble 1929). Thanks to the exquisite precision of today's observational data, we are able to test the theoretical predictions of cosmological models against independent data sets mapping the Universe at different times throughout its history and on a wide range of scales. Nevertheless, fundamental questions about the origin, the components and the dynamics of the Universe remain unresolved, making cosmology an exciting and active area of research.

A fundamental assumption of modern cosmology is the cosmological principle, borrowing from the Copernican idea that Earth does not occupy a privileged position in space. The cosmological principle states that on large scales, the Universe is *homogeneous*, i.e., it looks the same from any observing position in the Universe, and *isotropic*, i.e., it looks the same in every direction. A natural consequence of this is that our observable Universe is assumed to be a representative sample of the whole and the same laws of physics apply throughout. This simple yet powerful assumption legitimises the use of observations made from Earth to test cosmological models. Although initially a mathematical convenience, the cosmological principle has now been empirically confirmed; isotropy has been established by observations of the cosmic microwave background (CMB, Penzias and Wilson 1965), whereas large-scale galaxy surveys confirmed that the distribution of matter is homogeneous at scales larger than $\sim 100\,\mathrm{Mpc}$, up to the largest observable scales of $\sim 600\,\mathrm{Mpc}$ (Davis et al. 1982; Maddox et al. 1990; Percival et al. 2001).

1.1.1 General relativity

The fundamental theoretical bedrock of modern cosmology is the theory of general relativity[1]. Unlike Newtonian theory where gravity is an external force exerted on an object, in general relativity gravity is a geometric property of spacetime. The properties of any given spacetime can be mathematically expressed in terms of the *metric*, which relates coordinate distances to physical ones. The theory of general relativity connects the metric to the matter and energy content of the Universe via the Einstein field equations

$$G_{\mu\nu} \equiv R_{\mu\nu} - \frac{1}{2} g_{\mu\nu} R = 8\pi G T_{\mu\nu} + \Lambda g_{\mu\nu} \tag{1.1}$$

where, on the left-hand-side, $G_{\mu\nu}$ is the *Einstein tensor*, $R_{\mu\nu}$ and $R = g^{\mu\nu} R_{\mu\nu}$ are the *Ricci tensor* and *Ricci scalar* respectively, and $g_{\mu\nu}$ is the metric tensor. On the right-hand-side, $T_{\mu\nu}$ is the energy-momentum tensor, G is Newton's constant and Λ is the so-called cosmological constant. Setting aside the term involving the cosmological constant, Einstein's equations tell us that the geometry of the Universe, described by the Einstein tensor, is determined by its matter and energy content, described by the energy-momentum tensor. The cosmological constant was originally introduced by Einstein on the left-hand-side of Eq. (1.1), to allow for a static universe that is stable against gravitational collapse (Einstein 1917). Once Hubble observationally discovered the expansion of the Universe, Einstein abandoned the cosmological constant completely, calling it "the biggest blunder of his life". Today, the cosmological constant is generally seen on the right-hand-side of Eq. (1.1), representing a "vacuum energy" of spacetime itself, rather than of the matter content, which also contributes to the total energy of the Universe.

Under the assumption that the Universe is homogeneous and isotropic on large scales, the Einstein field equations for an expanding universe are solved by the Friedman-Lemaitre-Robertson-Walker (FLRW) metric of the form

$$ds^2 = -\mathrm{d}t^2 + a(t)^2 \left[\frac{\mathrm{d}r^2}{1 - kr^2} + r^2 \left(\mathrm{d}\theta^2 + \sin^2\theta \mathrm{d}\phi^2 \right) \right], \tag{1.2}$$

where k is the spatial curvature constant and takes only the discrete values $\{-1, 0, +1\}$ corresponding to open, flat and closed geometries, respectively. The scale factor $a(t)$ is a convenient parameter to describe the background expansion (or contraction) of the Universe as a function of time and conventionally, takes $a = 1$ today.

[1] See Dodelson (2003) for a good review of general relativity in the context of cosmology.

From the FLRW metric alone, one can derive several quantities which are useful for the description of an expanding universe. The time evolution of the scale factor can be expressed by the expansion rate of the Universe, also called the Hubble parameter,

$$H(t) = \frac{1}{a}\frac{\mathrm{d}a}{\mathrm{d}t}. \tag{1.3}$$

The value of $H(t)$ today, H_0, is called the Hubble constant and is usually expressed in terms of the dimensionless parameter h such that $H_0 = 100\,h\,\mathrm{km\,s^{-1}\,Mpc^{-1}}$.

A natural consequence of an expanding universe is that photons are redshifted due to space dilatation. Given that light rays travel on null-geodesics of spacetime i.e., $ds^2 = 0$, the maximum comoving distance travelled by a photon between time t_i and t is given by

$$\chi(t) = \int_{t_i}^{t} \frac{dt'}{a(t')} \equiv \eta, \tag{1.4}$$

where we have additionally defined the conformal time η. This defines a causal horizon, beyond which particles have not been causally connected since t_i. Moreover, the fact that the physical momentum and energy of photons are inversely proportional to the scale factor leads to the definition of a cosmological redshift in terms of the observed and emitted wavelengths, λ_{obs} ,and λ_{emit},

$$\frac{\lambda_{\mathrm{obs}}}{\lambda_{\mathrm{emit}}} \equiv 1 + z = \frac{a_{\mathrm{obs}}}{a_{\mathrm{emit}}}, \tag{1.5}$$

thus relating the wavelengths and scale factors at emission and observation times. For the case where the emission frequency of the photon can be determined (e.g. when the physical process is known), the redshift can be used as a distance indicator.

We now turn to the right-hand-side of the Einstein field equations (Eq. (1.1)), which involves the matter and energy content in the Universe. Matter in a homogeneous and isotropic cosmological background evolves as a perfect fluid, which by definition is completely characterised by its rest frame mass density ρ and isotropic pressure P, related via an equation of state $\rho = wP$. The energy-momentum tensor of a perfect fluid is given by $T_{\mu\nu} = \mathrm{diag}\,(-\rho,\, P,\, P,\, P)$. Inserting the FLRW metric and the energy-momentum tensor into Eq. (1.1), one obtains the two Friedmann

equations,

$$\left(\frac{\dot{a}}{a}\right)^2 = \frac{8\pi G}{3}\rho - \frac{\kappa}{a^2}, \tag{1.6}$$

$$\frac{\ddot{a}}{a} = -\frac{4\pi G}{3}(\rho + 3P) + \frac{\Lambda}{3}, \tag{1.7}$$

which describe the expansion rate and the acceleration of the Universe as a function of its density, pressure and spatial curvature. Imposing the additional constraint of energy-momentum conservation, $\nabla_\mu T^\mu_\nu = 0$, the Friedmann equations yield a third relation

$$\dot{\rho} + 3\frac{\dot{a}}{a}(\rho + P) = 0. \tag{1.8}$$

The solution to Eq. (1.8) is given by $\rho \propto a^{-3(1+w)}$ describes the evolution of the energy contents of the Universe as it expands according to the scale factor $a(t)$. This solution allows us to directly relate the evolution of the scale factor to the equation of state parameter of a given fluid (when plugged back into Eqs. 1.6 and 1.7):

$$a(t) = \begin{cases} t^{\frac{2}{3(1+w)}}, & \text{if } w \neq -1, \\ e^{Ht}, & \text{if } w = -1. \end{cases} \tag{1.9}$$

This result implies that we can describe the expansion of the Universe at any given time given the equation of state parameter w of the fluid which dominates the total energy density of the Universe at that time. A radiation-dominated universe has $a \propto t^{1/2}$ since relativistic particles have $w = 1/3$, whereas matter-domination implies $a \propto t^{2/3}$ since $w = 0$. A cosmological constant with $w = -1$ causes a exponential growth of the scale factor, provided H is constant. It is therefore also useful to express the abundance of any component in the Universe in units of the *critical density* ρ_c, defined as the energy density if the Universe is flat ($k = 0$),

$$\rho_c = \frac{3H^2}{8\pi G}. \tag{1.10}$$

Each component i of density ρ_i has an associated parameter $\Omega_i \equiv \rho/\rho_c$, such that the Friedmann equation Eq. (1.6) becomes

$$\Omega(a) - 1 = \frac{k^2}{H^2 a^2}, \tag{1.11}$$

where $\Omega(a)$ is the density parameter summing the energy density from all forms of constituents of the Universe. The various Ω_i evolve with time differently, depending on the equation of state of the component, which in turn will affect the expansion of the Universe. In the following sections, I will outline the components of the Universe according to our standard model of cosmology and their resulting effects on the evolution of the Universe.

1.1.2 The energy content

Einstein's equations explicitly reveal that the dynamics of the Universe are determined by its energy and matter content. In the current $\Lambda\mathrm{CDM}$ concordance model, the Universe is mainly composed of four ingredients:

1. *Baryonic matter*: making up only 5% of the total energy budget today, baryonic matter includes all atomic nuclei and electrons (even though the latter are leptons), interacting through gravitational, electromagnetic, strong and weak forces. It is predominantly made of hydrogen and light elements formed in the early Universe, composing galaxies, stars, planets, and all living organisms in our Universe.

2. *Relativistic species*: mainly consisting of relic electromagnetic radiation, the CMB (CMB), and neutrinos (CνB) produced in the early Universe. These make $< 1\%$ of today's total energy budget, but play a major role in the formation of structure at small scales.

3. *Cold dark matter*: the predominant component of the matter sector in the Universe, made of non-baryonic, pressureless and non-relativistic matter. Although at present its nature remains unknown, its existence has been confirmed indirectly based on its gravitational effects on various cosmological and astrophysical observations[2]: galaxy clusters (Oort 1932; Zwicky 1933), galaxy rotation curves (Freeman 1970; Rubin and Ford 1970), gravitational lensing (Massey et al. 2010), where a particular iconic example is the Bullet-Cluster (Clowe et al. 2006), and finally, the CMB (Planck Collaboration et al. 2018a). The first convincing observations for the existence of dark matter came in 1970 with the observations of galactic rotation curves; at the time, one of the leading explanations for the anomalous dynamics of these objects was the existence of a large population of dim astrophysical objects, known as massive compact halo objects (MACHOs). CMB observations, however, showed strong evidence that the dark matter could not be baryonic, excluding all but one MACHO candidate,

[2]For a good review on the history of dark matter see Bertone and Hooper (2018).

the primordial black hole (PBH; Carr and Hawking 1974). Observational constraints on PBHs today forbid them from accounting for the entirety of dark matter, unless they live near $M \sim 10^{-12}\,\mathrm{M}_\odot$ or $10^{-15}\,\mathrm{M}_\odot$ and have a monochromatic mass function[3] (see e.g. Sato-Polito et al. 2019). The simplest way to evade the constraint that dark matter be non-baryonic is to postulate that dark matter is comprised of a new non-baryonic particle, that interacts very weakly with the known particles of the Standard Model. A vast array of dark matter candidates have been proposed over the last decades to explain the existence of dark matter; the leading candidates have been weakly interacting massive particles (WIMPS; Steigman and Turner 1985), axions (Peccei and Quinn 1977; Weinberg 1978; Wilczek 1978) and sterile neutrinos (Dodelson and Widrow 1994).

4. *Dark energy*: another unknown component responsible for today's accelerated expansion of the Universe, which accounts for 69.4% of today's total energy content of the Universe. Its evidence came from observations of Type Ia supernovae (Perlmutter et al. 1999; Riess et al. 1998) by comparing their observed luminosity distance to that expected in a dark-matter-dominated universe and in a dark-energy-dominated one. In $\Lambda\mathrm{CDM}$, dark energy takes its simplest form of a cosmological constant Λ with a constant energy density ρ_Λ and equation of state $w = -1$. Although current observations are consistent with this model, taking Λ as the vacuum energy leads to the notorious *cosmological constant problem*; theoretical predictions from quantum field theory disagree with the observed value by 120 orders of magnitude. For this reason, alternative dark energy models, e.g. one with time-evolving equation of state $w(a) = w_0 + (1 - a(t))w_a$, or theories of modified gravity that lead to accelerated expansion are currently active areas of research in cosmology. See Martin (2012) and Huterer and Shafer (2018) for recent reviews on dark energy.

1.1.3 Expansion history

There are three key pieces of evidence which support the idea of an expanding universe: the Hubble-Lemaître law on the observed relation between distance and recession velocity of galaxies (Hubble 1929; Slipher 1915), the observed abundances of light elements in agreement with Big Bang Nucleosynthesis (BBN) predictions (Alpher et al. 1948), and observations of the CMB (Penzias and Wilson 1965). The fact that the Universe is expanding directly implies that the Universe was once smaller, denser and hotter. Fig. 1.1 shows a qualitative illustration of the history of the

[3] Generally, the allowed fraction of PBH dark matter decreases with increasing the width of the mass function. See Carr et al. (2017) for constraints on PBH dark matter with an extended mass function.

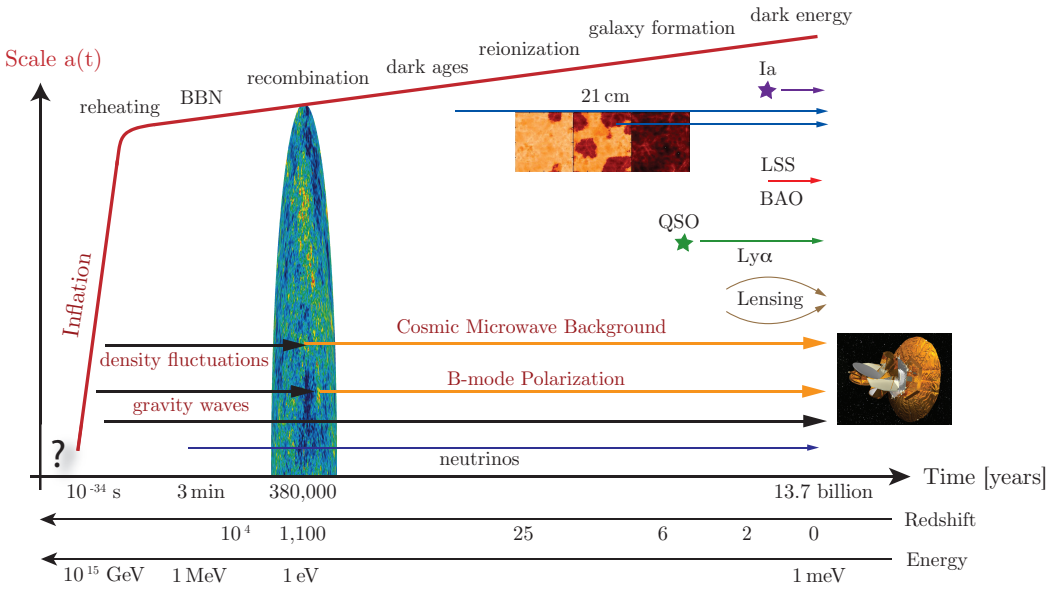

Figure 1.1: Outline of the key events in the history of the Universe as a function of time, redshift and energy scales, together with the cosmological probes used to study the different epochs in cosmic history. Figure taken from Baumann (2011).

Universe, together with some of the cosmological probes used to study different epochs of cosmic history.

In the standard cosmological paradigm, the history of the Universe begins with a phase of accelerated exponential expansion known as *inflation*, responsible for the homogeneity and isotropy of our observable Universe on large scales. During this expansion, small quantum fluctuations are turned into density perturbations of cosmological scales, providing the seeds for the formation of all cosmic structures in the Universe. At the end of inflation, the inflaton decays to produce all known particle species in the Standard Model during the epoch of *reheating*.

At this stage, electromagnetic radiation dominates the energy budget of the Universe, such that the scale factor grows as $a(t) \propto t^{1/2}$ and the expansion is decelerating. The energy contents can be described by an ionised hot plasma of baryons and radiation tightly coupled together and in thermal equilibrium. The first particles to decouple from the plasma as the temperature cools are neutrinos, forming a cosmic neutrino background yet to be detected. As the temperature drops further to a few MeV when the Universe is several minutes old, Big Bang nucleosynthesis begins, where protons and neutrons combine to produce higher elements (mainly deuterium, helium and lithium).

At around $z \sim 4800$, the mean photon energy density has decreased enough for the energy density of matter to exceed that of radiation. The Universe enters the so-called *matter-dominated*

era; matter perturbations grow linearly with the scale factor and cosmological structures start to form. At a temperature of about 3000 K, when the Universe was about 380,000 years old, electrons and protons combine into neutral atoms, while photons are released as a homogeneous background radiation known as the CMB. At this point, baryons are no longer prevented from clustering due to photon pressure, but fall into the potential wells created by the dark matter. Perturbations continue to grow during the *dark ages*, where the predominant dark matter collapses into halo-like structures through its own gravitational attraction. Eventually, the highest dark matter overdensities contain enough neutral hydrogen at sufficiently high densities to form the first stars and galaxies. *Reionization* subsequently takes place once the first stars and galaxies have formed, and their radiation is energetic enough to re-ionize the neutral hydrogen in the surrounding intergalactic medium. The process of reionization is though to start with ionized bubbles around the strongest ionizing sources, which slowly grow to fill the Universe with an ionized plasma. This epoch is still poorly constrained but will soon be refined by measurements of the 21-cm transition, mapping the distribution of neutral hydrogen at the epoch of reionisation.

After approximately 10 billion years, the Universe's expansion starts to accelerate as dark energy overtakes matter and becomes the dominant component in the Universe.

1.2 Seeds of cosmic structures

A universe uniquely defined by an FLRW metric could not have formed the complex large-scale structure that we observe in our Universe. If the matter distribution was simply homogeneous and isotropic on all scales, it would remain to be so throughout cosmic history. Instead, galaxy surveys starting in the 1970s, such as the Lick galaxy catalogue (Seldner et al. 1977), the CfA Redshift Survey (Davis et al. 1982) and the APM Galaxy Survey (Maddox et al. 1990), revealed that matter is arranged into a well-defined *cosmic web*, with galaxies located in filaments and at their intersections, with huge empty voids surrounding them. Our model of the Universe must therefore allow for the existence of initial perturbations over an otherwise homogeneous background described by an FLRW metric, forming the seeds of all cosmic structures. The CMB provides us with the earliest picture of the inhomogeneous Universe, showing tiny temperature perturbations which trace the underlying initial density perturbations. In this section, I will discuss how the CMB determines the initial conditions of structure growth and how the theory of inflation can be used to explain their origin.

1.2.1 The cosmic microwave background

At the time of recombination of electrons and protons when the Universe cooled below $\sim 3000\,\mathrm{K}$, photon decoupling led to the emission of a relic radiation known as the CMB. It was first observed by Penzias and Wilson (1965) and later identified by Dicke et al. (1965). The existence of this background radiation provides direct evidence that the Universe was once hot and dense as predicted by the hot Big Bang theory. The first measurements were made by the FIRAS experiment aboard the COBE satellite in the 1990s (Mather et al. 1994), which measured its near-perfect blackbody spectrum with a temperature of $2.7\,\mathrm{K}$. In addition to these, COBE also measured small temperature fluctuations in the CMB of the order of 1 in $100,000$ (Smoot et al. 1992). With its resolution of 7 degrees on the sky, the COBE satellite could only see the largest angle fluctuations, capturing information about the initial conditions of the Universe. Subsequent satellite missions such as *WMAP* (Bennett et al. 2003) and *Planck* (Planck Collaboration et al. 2018a) enabled the study of the temperature fluctuations in the CMB down to angular scales smaller than a tenth of a degree, and were further complemented by high-resolution ground-based experiments, such as the Atacama Cosmology Telescope (ACT) (Das et al. 2014) and the South Pole Telescope (SPT) (George et al. 2015). These high-precision measurements led to the establishment of the $\Lambda\mathrm{CDM}$ model, already coined the "concordance" model by Ostriker and Steinhardt (1995), as the standard model of cosmology and to an era of precision cosmology, where quantitative predictions about the origin of structure and the content of matter and energy in the Universe can be tested against observations.

Figure 1.2 shows measurements of the CMB angular power spectra from a variety of different experiments, compared to the prediction of the $\Lambda\mathrm{CDM}$ model with the *Planck* best-fit cosmological parameters (dashed line). The upper panel shows the power spectra of the temperature and E-mode and B-mode polarization signals, the middle panel the cross-correlation spectrum between T and E, while the lower panel shows the CMB lensing power spectrum. The data and the model are in excellent agreement over a wide range of scales. The amplitude and features of the power spectra contain information about the geometry and composition of the Universe, together with its dynamics before and after recombination.

Remarkably, observations of the CMB can be described by a spatially-flat $\Lambda\mathrm{CDM}$ model with only six free paramaters; no evidence is found for extensions of this model that can provide a better fit to observations (Planck Collaboration et al. 2018a). Two of the six cosmological parameters describe how the energy density distributed in the three components[4] – dark matter, ordinary matter and

[4]Note that only two parameters are needed under the assumption that the Universe is flat i.e., $\Omega = 1$.

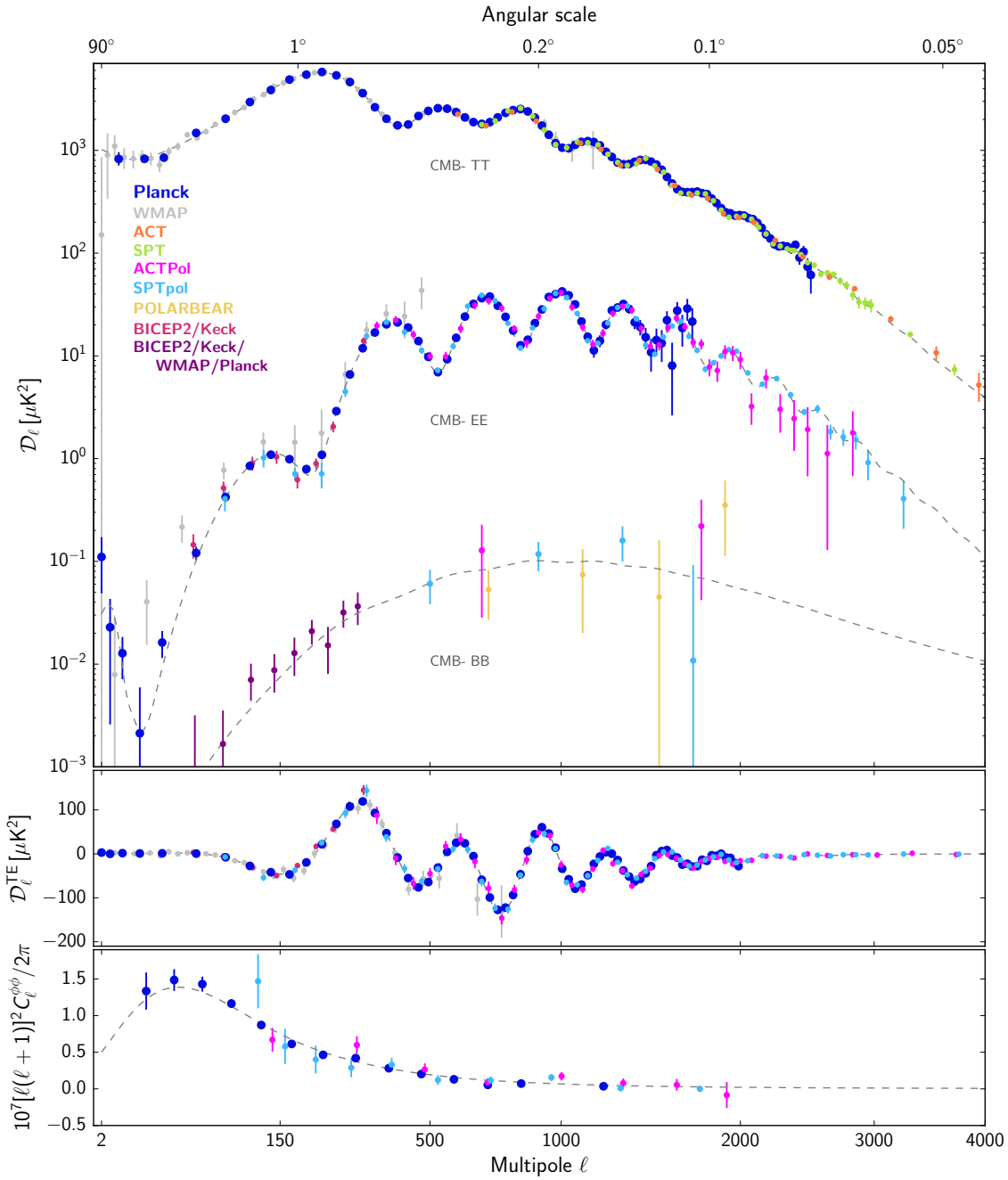

Figure 1.2: Compilation of measurements of the CMB angular power spectra. The upper panel shows the power spectra of the temperature and E-mode and B-mode polarization signals, the next panel the cross-correlation spectrum between T and E, while the lower panel shows the lensing deflection power spectrum. Different colours correspond to different experiments, and the dashed line shows the best-fit ΛCDM model to the *Planck* temperature, polarization, and lensing data. Figure taken from (Planck Collaboration et al. 2018b).

dark energy – and are parametrized as $\Omega_c h^2$ and $\Omega_b h^2$. Another two describe the initial conditions of the Universe: the amplitude A_s and power-law scale index n_s of the primordial curvature power spectrum from inflation (see Sec. 1.2.2). Finally, the remaining two free parameters relate to the phase transitions of recombination and reionization: one is the angular size of the sound horizon at recombination, $\theta_\star$, and the other is the optical depth, τ, due to Thomson scattering at the time of reionization. The best-fit values for the cosmological parameters obtained from the final *Planck* data release are: dark matter density $\Omega_c h^2 = 0.120 \pm 0.001$, baryon density $\Omega_b h^2 = 0.0224 \pm 0.0001$, scalar spectral index $n_s = 0.965 \pm 0.004$, optical depth $\tau = 0.054 \pm 0.007$, angular acoustic scale $100\theta_* = 1.0411 \pm 0.0003$ and amplitude $\ln(10^{10} A_s) = 3.044 \pm 0.014$ at $68\,\%$ confidence level.

Assuming the best-fit ΛCDM cosmology, one can then infer late-Universe parameters such as the Hubble constant $H_0 = (67.4 \pm 0.5)\,\mathrm{km/s/Mpc}$, matter density parameter $\Omega_m = 0.315 \pm 0.007$ and the root-mean-squared (rms) of matter fluctuations in $r = 8\,h^{-1}\,\mathrm{Mpc}$ spheres, $\sigma_8 = 0.811 \pm 0.006$. The choice of $r = 8\,h^{-1}\,\mathrm{Mpc}$ comes from early measurements of the two-point correlation function from the Lick galaxy catalogue, where the variance of galaxy counts was found to be roughly unity within a radius of $8\,h^{-1}\,\mathrm{Mpc}$ (Peebles 1980). Perhaps the most interesting of cosmological parameters at present day is the Hubble constant H_0. Direct measurements of H_0 from observations of Type Ia supernovae (SN Ia), based on distances calibrated to Cepheid variables, yield a value $H_0 = 74.03 \pm 1.42\,\mathrm{km/s/Mpc}$, which is $4.4\,\sigma$ discrepant with the value inferred by *Planck* (Riess et al. 2019). Future measurements from independent data, such as local SN Ia measurements calibrated to tip of the red giant branch stars (Freedman et al. 2019) and gravitational waves (Feeney et al. 2019; Soares-Santos et al. 2019), may resolve the tension by revealing whether this discrepancy indicates new physics beyond ΛCDM or unaccounted systematic uncertainties in the measurements.

1.2.2 Inflation

The CMB provides us with the earliest possible picture of our Universe, showing that the early Universe was extremely homogeneous and isotropic (up to $\sim 10^5$ accuracy) on all scales. This naturally gives rise to the question of what sort of initial conditions lead to such homogeneity and isotropy. In the standard picture of a hot Big Bang cosmology described by general relativity, the initial conditions represent a singularity where the theory breaks down and the initial conditions cannot be imposed. This singularity problem indicates the need of a quantum theory of gravity to describe the Universe at its creation time.

A different solution for setting the initial conditions within the context of general relativity comes with the name of *cosmic inflation* and was first proposed by Guth (1981) to solve a number of problems that otherwise arise in the ΛCDM model. Inflation not only provides an elegant solution to weaknesses in the standard hot Big Bang model but also leads to quantitative (and testable) predictions for the origin of structure in the Universe. In this section, I will review the problems in the ΛCDM model which inflation is intended to resolve and its mechanism for the generation of primordial fluctuations.

The first problem is related to the homogeneity observed in the CMB at scales larger than the Hubble radius at the time of CMB decoupling. It is known as the *horizon problem*: regions of the sky separated by distances larger than the Hubble radius are not in causal contact but nevertheless appear homogeneous. It is useful to rewrite the comoving horizon η in terms of the comoving Hubble radius $(aH)^{-1}$,

$$\eta(a) = \int_0^a \frac{da'}{a'} \frac{1}{a'H(a')}. \tag{1.12}$$

The particle horizon represents the scale on which particles have *ever* been able to interact since $t = 0$, whereas the Hubble radius sets the scale of causality at any given time. Inflation resolves the horizon problem by allowing particles separated by any given scale to have been in causal contact at some earlier time. This can be achieved if during the inflationary epoch the comoving Hubble radius *decreases* with time. At the start of inflation, the Hubble radius was so large that all scales were well within the horizon and therefore causally interacting. At that time, these regions were given the necessary initial conditions and the smoothness we observe today in the CMB. During inflation, scales eventually fell out of contact and re-entered the horizon only at later times during standard expansion.

The second problem arises from the fact that the Universe is observed to be consistent with a flat geometry or equivalently, that today's total energy density in the Universe is very close to the critical density i.e., $\Omega \sim 1$. A closer inspection of Eq. (1.11) reveals that the near-flatness observed today requires the fine-tuning of Ω to an even closer value to 1 in the early universe. This is because the comoving Hubble radius, $(aH)^{-1}$, grows with time and hence any deviation from perfect flatness implies that the quantity $|\Omega - 1|$ diverges with time. This fine-tuning problem, also known as the *flatness problem*, can be resolved by inflation; if the comoving Hubble radius decreases with time, Ω naturally tends to 1 at the end of inflation.

In general, the key feature of the inflationary epoch is that of a decreasing comoving Hubble

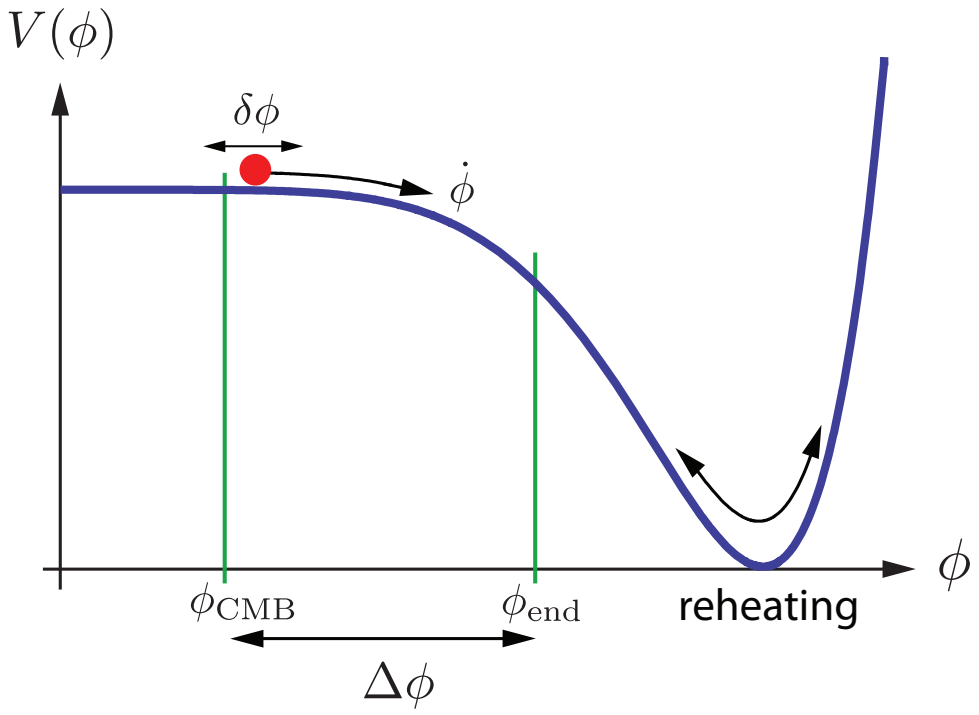

Figure 1.3: Illustration of slow-roll inflation; the inflaton slowly rolls along the shallow slope of the potential whilst $\dot{\phi} \ll V(\phi)$. During that time, local quantum fluctuations $\delta\phi(x,t)$ are also present around its mean value $\phi(t)$. At the end of inflation, the field oscillates around the potential's minimum and "reheats" the Universe. Figure taken from Baumann (2011).

radius. This directly implies $\mathrm{d}^2a/\mathrm{d}t^2 > 0$, i.e. an accelerated expansion. For inflation to solve the horizon problem, the comoving Hubble radius at the start of inflation must also be larger than the largest scales observable today (today's comoving Hubble radius). If we assume H is approximately constant during inflation, the comoving Hubble radius must decrease by at least 28 orders of magnitude, meaning that the scale factor must increase by at least a factor of $\sim e^{60}$ ($60\,e$-folds). The accelerated expansion during inflation is usually attributed to a scalar field ϕ, known as the *inflaton*, which can be described by a fluid of negative pressure with an equation of state $w < -1/3$. In the simplest inflationary scenario, the inflaton slowly rolls down a near-flat potential as illustrated in Fig. 1.3 (Albrecht and Steinhardt 1982; Linde 1982); this "slow-roll" regime can be achieved provided the kinetic energy of the field is much smaller than its potential energy. Once the kinetic energy becomes comparable to the potential energy, inflation ends and the field oscillates around the potential's minimum thus entering the epoch of reheating (see Sec. 1.1.3; Mukhanov and Chibisov 1981).

Although inflation was first introduced to solve problems within the standard Big Bang model, it

also provides a theory that explains the origin of the density fluctuations and predicts their power spectrum. Although the details of the fluctuations depend on the specific inflationary models, CMB observations provide a stringent test to some of the general features predicted by inflation. One of these is that the inflaton field has local quantum fluctuations $\delta\phi(x,t)$ around its mean value $\phi(t)$ (Guth and Pi 1982; Linde 1982; Mukhanov and Chibisov 1981; Starobinsky 1982), as illustrated in Fig. 1.3. These fluctuations translate into a locally different time evolution, where each patch evolves faster or slower depending on the sign of the fluctuation. This creates local density perturbations, which become classical as the Hubble volume shrinks and the size of the perturbations becomes larger than the horizon. The density perturbations are sourced by the comoving adiabatic curvature perturbations $\mathcal{R}(\mathbf{x},t)$, and their amplitude is (almost) unchanged from the time of horizon crossing. The curvature perturbations are Gaussian and can therefore be entirely characterized by their (dimensionless) power spectrum

$$\Delta^2_{\mathcal{R}}(k) = \left.\frac{1}{8\pi^2}\frac{H^2}{M^2_{\rm pl}}\frac{1}{\epsilon}\right|_{k=aH} \tag{1.13}$$

where $M_{\rm pl} = (8\pi G)^{-1/2}$ is the Planck mass, ϵ is a slow-roll parameter which depends on the Hubble rate and the dimensionless power spectrum is defined as $\Delta^2(k) = \left(k^3/(2\pi)^3\right) P(k)$.

$\Delta^2_{\mathcal{R}}(k)$ is usually parametrised as $\Delta^2_{\mathcal{R}}(k) = A_s\,(k/k_0)^{n_s-1}$, where A_s is the amplitude, n_s is the scalar spectral index and k_0 is an arbitrary pivot scale. The value of the scalar spectral index inferred by Planck Collaboration et al. (2018a) is $n_s = 0.9649 \pm 0.004$ at 68% confidence level, confirming the nearly scale-invariant power spectrum predicted by inflation. In particular, the deviation of n_s from unity, which introduces a small scale-dependence in the power spectrum expected from inflation, has been observationally confirmed to an accuracy of 8.4σ, providing strong evidence for the slow-roll inflationary paradigm (Planck Collaboration et al. 2018a).

1.3 Cosmological structure formation

Gravitational instability is predominantly responsible for the growth of structure in the Universe. Initially overdense regions accumulate more and more matter as time evolves, eventually producing the non-linear structure we observe today. In a ΛCDM universe, structure growth is hierarchical; smaller objects form first and subsequently merge to form even larger structures. Large-scale structures mainly originate from perturbations to the dark matter, which will be the focus of this section. Although in principle these are coupled to all other perturbations, they depend on radiation

perturbations only indirectly via the gravitational potential. In fact, during the radiation-dominated era, the potential is determined by radiation perturbations, whereas dark matter perturbations are influenced by the behaviour of the potential but do not influence the potential themselves. As the Universe enters the matter-dominated era, matter perturbations will instead drive the evolution of the potential and grow accordingly. In addition to this, the evolution of cosmological perturbations at any given time depends on the size of the wavelength modes; super-horizon and sub-horizon modes i.e., perturbations of size larger or smaller than the Hubble radius, evolve differently.

It is useful to define the density contrast, δ, as

$$\delta = \frac{\rho - \bar{\rho}}{\bar{\rho}}, \tag{1.14}$$

where $\bar{\rho}$ is the mean density of the Universe at a given redshift. During the expansion of the Universe, the density contrast is affected by three main factors: amplification due to gravitational instability, pressure and dissipation (Dodelson 2003). The interplay between gravity and pressure dictates whether fluctuations grow driven by gravity or oscillate in time if driven by pressure. The critical scale for which gravity and pressure find equilibrium is known as the *Jeans length*. In particular, for an expanding universe, fluctuations below the Jeans' scale oscillate with decreasing amplitude and fluctuations above the Jeans scale experience power-law growth. The Jeans length is

$$\lambda_J = c_s(t)\sqrt{\frac{\pi}{G\bar{\rho}(t)}}, \tag{1.15}$$

where the sound speed $c_s(t)$ and $\bar{\rho}(t)$ both depend on time.

Although structure evolution is predominantly non-linear, the equations describing the evolution of structure can be linearised if the density fluctuations are small. We can then make use of linear perturbation theory to find explicit analytical solutions. The initial perturbations were indeed small, as demonstrated by the smallness of the CMB anisotropies, and therefore the results from linear theory are relevant to the onset of structure formation.

1.3.1 Linear growth

The correct description of linear-order structure evolution requires general relativistic perturbation theory. Consider the metric of spacetime as a sum of a homogeneous background metric $\bar{g}_{\mu\nu}$ and a

perturbation $\delta g_{\mu\nu}$ such that

$$ds^2 = (\bar{g}_{\mu\nu} + \delta g_{\mu\nu})dx^\mu dx^\nu, \tag{1.16}$$

where $\bar{g}_{\mu\nu}$ solves Einstein's equations in a homogeneous universe and $\delta g_{\mu\nu} \ll 1$.

In relativistic perturbation theory there is no obvious choice of coordinate system. The problem is that relativistic perturbation theory contains gauge freedom, meaning that the metric perturbations are not uniquely defined. Different choices of coordinates or "gauge choices" for the metric imply the same observable gauge-independent predictions (Ma and Bertschinger 1995). However, this freedom in coordinate choice leads to fictitious perturbation modes which reflect the gauge choice used and not real inhomogeneities.

One particularly convenient gauge choice, or gauge-fixing, is known as *conformal Newtonian gauge*, which will nicely connect GR to Newtonian theory in the case of scalar perturbations. The metric in conformal Newtonian gauge takes the form

$$ds^2 = a^2(\eta)\left[(1+2\Phi)d\eta^2 - (1-2\Psi)\delta_{ij}dx^i dx^j\right], \tag{1.17}$$

where η is conformal time (as defined in Eq. (1.4)) and the two potentials Ψ and Φ are the two new degrees of freedom.

As mentioned before, matter in a homogeneous and isotropic universe takes the form of a perfect fluid, with energy-momentum tensor $T^{\mu\nu} = (\rho + P)U^\mu U^\nu - Pg^{\mu\nu}$ and equation of state, $P = w\rho$. Solving Einstein's equations using the metric in Eq. (1.17) and a perfect fluid's energy-momentum tensor, one finds $\Psi = \Phi$; the evolution of the perturbations are actually affected by a single Newtonian-like potential. As a result, the equation of structure growth becomes that for a single potential,

$$\Phi'' + 3(1+w)\mathcal{H}\Phi' + wk^2\Phi = 0, \tag{1.18}$$

where $'$ denotes the derivative with respect to conformal time η and we have defined the conformal Hubble parameter $\mathcal{H} = a'/a = aH$. The evolution of density perturbations are then related to the potential via Poisson's equation,[5] which in Fourier space takes the form $-k^2\Phi = 4\pi G a^2 \delta\bar{\rho}$.

In the radiation-dominated era, the Universe is dominated by electromagnetic radiation with

[5]Technically, Poisson's equation in GR contains an extra term. However, for density perturbations in the *synchronous gauge* the relation between density perturbations and potential simplifies to the Newtonian form of Poisson equation. We therefore make the assumption $\delta\rho = \delta\rho_{\text{synchronous}}$.

an equation of state parameter $w = 1/3$, and the scale factor evolves as $a(t) \propto t^{1/2}$. It follows that $\mathcal{H} = 1/\eta$ and Eq. (1.18) becomes

$$\Phi'' + \frac{4}{\eta}\Phi' + \frac{k^2}{3} = 0. \tag{1.19}$$

The solution to Eq. (1.19) depends on the modes' wavelength; the gravitational potential is approximately constant when the modes are outside the horizon, whereas for modes inside the horizon, the potential oscillates with decreasing amplitude. Using Poisson equations to relate density perturbations to the potential, we find that the radiation overdensity evolves as $\delta_r \propto \eta^2\Phi$. During this time, structure growth is suppressed on sub-horizon scales by the dominant component (radiation) driving the potential fluctuations. This behaviour is known as the *Meszaros effect*; the growing mode of matter fluctuations scales logarithmically such that $\delta_m \propto \ln a$. On super-horizon scales instead, matter overdensities trace the behaviour of radiation such that $\delta_m \propto \delta_r \propto \eta^2$. This has the important consequence that the amplitude of a smaller perturbation will be suppressed by a factor of $(a_{\rm enter}/a_{\rm eq})^2$, where $a_{\rm enter}$ is the value of the scale factor at the time when the length scale of the perturbation is equal to the comoving horizon scale and $a_{\rm eq}$ is the value of the scale factor at the time of radiation-matter equality, due to the fact that fluctuations grow slower during the radiation-era if inside the horizon.

In the matter-dominated era, where $w = 0$ and $\mathcal{H} = 2/\eta$, the evolution of the potential takes the form

$$\Phi'' + \frac{6}{\eta}\Phi' = 0. \tag{1.20}$$

The growing solution is a constant Φ, meaning that it is frozen at all scales. From this, it follows that the growing mode of the density contrast grows proportionally to the scale factor such that $\delta_m \propto a$. The perturbations will then stop growing once the Universe enters the Λ-dominated era.

Most of the information in the matter fluctuations in the linear regime can be summarised in a statistical measure describing the power of fluctuations as a function of scale, namely the *matter power spectrum*. The power spectrum is the main cosmological observable of large-volume galaxy surveys and can be compared to theoretical expectations by assuming that galaxies are biased tracers of the underlying dark matter distribution. The latter is given by $P(k) \propto T(k)^2 P_0(k)$, where $T(k)$ is the *transfer function*, encapsulating the k-dependence and time-dependence of the evolution of perturbations from the end of slow-roll inflation to the present day, and $P_0(k)$ is the power

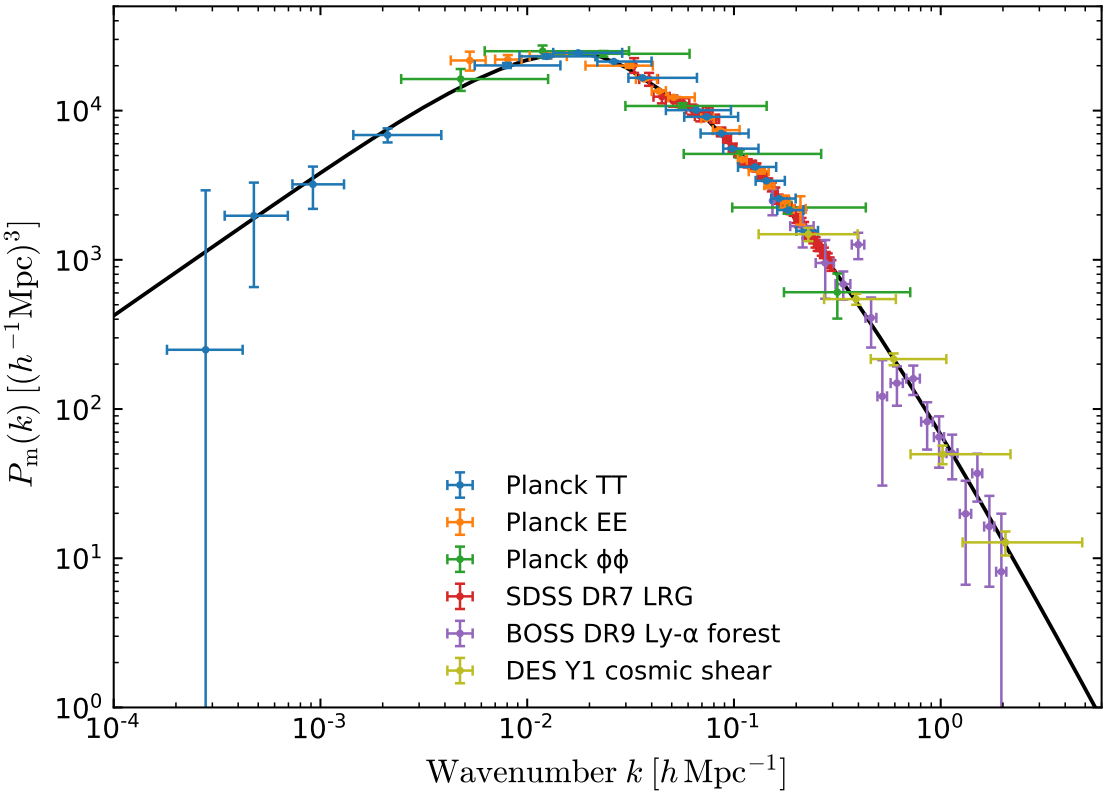

Figure 1.4: Linear matter power spectrum at $z = 0$ predicted by the ΛCDM model with the *Planck* best-fit cosmological parameters, compared to measurements from the CMB (Planck Collaboration et al. 2018a), galaxy clustering (Oka et al. 2014), the Lyman-alpha forest (Anderson et al. 2014) and weak lensing cosmic shear (Troxel et al. 2018). The model, fitted to the *Planck* data, agrees remarkably well with independent datasets, probing a wide range of spatial scales and different epochs of cosmic history.

spectrum of primordial fluctuations at the start of the radiation-dominated era.

The linear power spectrum at $z = 0$ predicted by the ΛCDM model is shown in Fig. 1.4, together with a compilation of measurements from the CMB (Planck Collaboration et al. 2018a), galaxy clustering (Oka et al. 2014), the Lyman-alpha forest (Anderson et al. 2014) and weak lensing cosmic shear (Troxel et al. 2018). The features of the linear power spectrum reflect the behaviour of matter fluctuations throughout cosmic history. At large scales, the power spectrum is basically given by the primordial power spectrum $P(k) \propto k$, whereas on small scales it is affected by the evolution of perturbations on sub-horizon scales, i.e. the Meszaros effect, so that $P(k) \propto k^{-3} \ln^2(k/k_{\rm eq})$. The separation between these two regimes originates from the suppression of the growth of the perturbations during the radiation-dominated era compared to the matter-dominated one, which

introduces a distinct length scale into the linear power spectrum and is given by the size of the horizon at radiation-matter equality. The other significant feature of the matter distribution is the imprint from acoustic oscillations present in the primordial baryon-photon plasma, known as *baryonic acoustic oscillations* (BAO), at the scale of the sound horizon at recombination. This results into wiggles in the power spectrum at $k \sim 0.04\,\mathrm{Mpc}^{-1}$.

The distribution of matter is not directly observable but can be inferred by tracers of the light distribution, e.g. galaxies and quasars, which probe the underlying large-scale structure through their positions, and the magnification and lensing of their light by gravity. Historically, galaxy surveys enabled detailed reconstructions of the large-scale structure of the Universe. Photometric surveys measure the spectrum of light only in a few broad band filters and use that to estimate the redshift. This method allows us to detect a large number of sources over large areas of the sky, therefore yielding large volumes and high statistical power. Examples of large three-dimensional galaxy surveys are: the Sloan Digital Sky Survey[6] (SDSS), the Dark Energy Survey[7] (DES), the largest galaxy survey to date, and in the future, Euclid[8] and the Large Synoptic Survey Telescope [9] (LSST). On the other hand, spectroscopic surveys such as Baryon Oscillation Spectroscopic Survery[10] (BOSS) and the Dark Energy Spectroscopic Instrument[11] (DESI), measure the full spectra of the sources, yielding more accurate measurements of redshifts for fewer objects. Fig. 1.4 also shows the inferred matter distribution inferred from other tracers: the distribution of neutral hydrogen probed by the absorption lines in quasar spectra of the Lyman-α transition (Ly-α forest; Anderson et al. 2014) and the weak gravitational lensing of galaxies, whose shapes are distorted and light is magnified by intervening matter between us and the sources (cosmic shear; Troxel et al. 2018).

The overlap of the different measurements with the theoretical predictions shown in Fig. 1.4 demonstrates the predictive power of the $\Lambda\mathrm{CDM}$ model and the success of the paradigm of structure formation in the linear regime. At small-scales, the power spectrum is affected by non-linear gravitational collapse and baryonic physics, which enhance the total power compared to its linear contribution. In this regime, one must resort to numerical simulations or higher-order perturbation theory.

[6] http://www.sdss.org
[7] http://www.darkenergysurvey.org
[8] https://www.euclid-ec.org
[9] http://www.lsst.org/lsst/
[10] http://www.sdss3.org/surveys/boss.php
[11] http://desi.lbl.gov

1.3.2 Non-linear growth

As the density fluctuations become non-linear i.e., $\delta > 1$, linear perturbation theory breaks down. Modes are no longer independent as they start to couple to each other due to gravity, meaning that the perturbations' evolution can no longer be described simply by the growth factor $D(a)$. Although no fully analytic solution exists, a number of approaches are possible in order to gain insights into the evolution of dark matter perturbations beyond the linear regime. Higher order perturbation theory, such as effective theories of large-scale structure, are useful to explain the quasi-linear regime, $\delta \sim 1$, whereas idealized approximations can provide useful physical insights well into the non-linear regime albeit their simplifications.

The evolution of a spherical mass overdensity is a special case of non-linear density evolution which finds an explicit analytical answer and it is known as the *spherical collapse model*. This model is useful in providing qualitative insights into the (much more complicated) process of non-linear collapse. A large fraction of the dark matter today resides in *dark matter haloes*; these form the building block of cosmic large-scale structure and it is within their potential wells that galaxies form. Therefore, it is vital to develop a physical understanding of their evolution and formation in order to understand their relation to galaxies and make predictions on the resulting galaxy distribution. Despite its assumptions and simplifications, the spherical collapse also provides powerful statistical predictions on the final structure of the Universe from properties of the initial conditions.

1.3.3 Spherical collapse model

The spherical collapse model provides an explicit solution to the non-linear evolution of density[12]. The main assumptions which characterize this model are that overdense regions are spherically symmetric and that spherical shells do not cross as they expand and collapse.

Consider a spherical shell of radius $r(t)$ containing mass M. The equation of motion in an EdS universe is

$$\frac{\partial^2 r}{\partial t^2} = -\frac{GM}{r^2}. \tag{1.21}$$

Using the parametric solutions $t(\theta) = B\,(\theta - \sin\theta)$ and $r(\theta) = A\,(1 - \cos\theta)$, one finds the relation $A^3/B^2 = GM$. At $t = 0$, or $\theta = 0$, the shell starts expanding until it reaches its maximum radius at $\theta = \pi$, such that $r = r_{\rm max} = 2A$ and $t = t_{\rm max} = \pi B$. After that, the shell recollapses to a point

[12]For a good review on the spherical collapse model see Mo et al. (2010).

at $\theta = 2\pi$, or $t_{\rm coll} = 2\pi B$. In reality, the shell will not collapse down to a point but to half of $r_{\rm max}$, called virial radius $r_{\rm vir}$, forming a virialised dark matter halo. The virial radius is a useful quantity as it defines a region which is found to describe well the collapsed part of the halo.

At early times, i.e. for small t and small θ, $t(\theta)$ can be written as a power series in θ (up to $5{\rm th}$ order), such that

$$t \approx B\left(\theta - \theta + \frac{\theta^3}{6} - \frac{\theta^5}{120}\right) \approx \frac{B\theta^3}{6}\left(1 - \frac{\theta^2}{20}\right) \implies \theta^3 \approx \frac{6t}{B}\left(1 - \frac{\theta^2}{20}\right)^{-1}, \tag{1.22}$$

where we have recovered an expression for θ as a function of t. Equivalently, $r(t)$ can be expanded as a power series in θ:

$$r \approx A\left(1 - 1 + \frac{\theta^2}{2} - \frac{\theta^4}{24}\right) \approx \frac{1}{2}\left(GM\right)^{1/3}\left(6t\right)^{2/3}\left[1 - \frac{1}{20}\left(\frac{6t}{B}\right)^{2/3}\right] \equiv r_{\rm EdS} - \Delta r, \tag{1.23}$$

where we used the expression for θ in Eq. (1.22), the relation $A^3 = GMB^2$ and the fact that in an Einstein-de Sitter (EdS) universe ($\Omega_m = 1$),

$$r = r_{\rm EdS} = \frac{1}{2}(GM)^{1/3}(6t)^{2/3}. \tag{1.24}$$

The assumption of an EdS universe dominated by cold dark matter and without a cosmological constant is a valid approximation during the matter-dominated era and it is therefore often used to simplify analytical calculations. Since $\rho \sim M/R^3$, the fractional overdensity of the sphere, compared to that of EdS, is

$$\delta = \frac{\Delta\rho}{\rho} \sim \left(-3\frac{\Delta R}{R^4}\right)R^3 \sim -3\frac{\Delta R}{R} = \frac{3}{20}\left(\frac{6t}{B}\right)^{2/3}. \tag{1.25}$$

Now that we have an explicit expression for the density contrast as a function of time in Eq. (1.25), we can follow the evolution of the sphere in the non-linear regime. In particular, for an EdS universe, the shell reaches its maximum expansion at $t = \pi B$. Therefore the density contrast at this turnover point is given by

$$\delta_{\rm lin} = \frac{3}{20}(6\pi)^{2/3} \simeq 1.06. \tag{1.26}$$

The shell then collapses to $r = 0$ at $t = 2\pi B$ and the density contrast predicted by linear theory is

$$\delta_{\rm lin} = \frac{3}{20}(12\pi)^{2/3} \simeq 1.69. \tag{1.27}$$

This result tells us that a spherical region of matter collapses at the present epoch when its linear density contrast exceeds $\delta_{\rm lin} \simeq 1.69$. Consequently, in order for the collapse to happen at a scale factor a, the linear density contrast must exceed $\delta_{0,\rm lin} = 1.69/D(a)$, where $D(a)$ is the growth factor. In a realistic scenario where a spherical region will only collapse to its virial radius, one finds that the overdensity of the sphere after virialisation is $\rho = 178\bar{\rho}$, using the virial theorem $E_{\rm kin} = -E_{\rm pot}/2$ at virial equilibrium. The result in Eq. (1.27) is valid under the assumption of an EdS universe; for cosmologies with $\Omega_m \neq 1$ and a non-zero cosmological constant, the linear collapse density threshold becomes $\delta_{\rm lin} \approx 1.69\,[\Omega_m]^{0.0055}$, to better than 1% accuracy (Lahav et al. 1991; Mo et al. 2010). Since the dependence on Ω_m is extremely weak, to good approximation $\delta_{\rm lin} \simeq 1.69$ for all realistic cosmologies.

The spherical collapse model provides useful results despite its strong assumptions, as we will see in the next sections. This model is far from what happens in reality; haloes do not simply grow and re-collapse to form virialized objects but rather form hierarchically as a result of continuous mergers and accretion events. Nevertheless, it carries interesting statistical implications which turn out to be in qualitative agreement with those found by N-body simulations.

1.3.4 Press-Schechter Formalism

The spherical collapse model provides simple analytic solutions to the equations of structure evolution. This model can be used to obtain statistical measures on the collapsed haloes, as for example the abundance of haloes as a function of mass and time, by relating the mass and collapse time of a halo to the linear density field. This was introduced for the first time by Press and Schechter (1974) and it is known as Press-Schechter (PS) theory. Press and Schechter (1974) were also the first to develop a cosmological simulation of structure formation with N-body integrations.

PS theory is based on the assumption that the fraction of volume collapsed in haloes of mass M is equivalent to the portion of initial density field smoothed on a mass scale M (or, equivalently radius scale R) which exceeds the density threshold $\delta_{\rm th}$. The latter is that derived from the spherical collapse model at $z = 0$, $\delta_{\rm th} = \delta_{\rm sc} \simeq 1.69$. Therefore, by counting the regions where the density contrast exceeds the density threshold, one can predict the number density of haloes as a function of mass.

Consider the linear density field at redshift z smoothed using a top-hat filter function at mass scale M. Assuming the smoothed density field δ_M is a Gaussian random field, the probability of the smoothed linear density field exceeding the density threshold $\delta_{\rm sc}$ at redshift z is

$$p_{>\delta_{\rm sc}}(M,z) = \frac{1}{\sqrt{2\pi}\sigma(M,z)} \int_{\delta_{\rm sc}}^{\infty} \exp\left(-\frac{\delta_M^2}{2\sigma^2(M,z)}\right) \mathrm{d}\delta_M, \tag{1.28}$$

where the mass variance is defined as the variance of the smoothed density field, given by

$$\sigma^2(M,z) = \int \frac{\mathrm{d}^3k}{(2\pi)^3} \left|\tilde{W}_M(k)\right|^2 P(k,z), \tag{1.29}$$

where $P(k,z)$ is the power spectrum and $\tilde{W}_M(k)$ is the Fourier transform of a spherical top-hat window function.

According to the PS ansatz, the probability in Eq. (1.28) is equal to the fraction of mass contained in haloes with mass greater than M. However, as $M \to 0$, then $\sigma(M) \to \infty$ and $p_{>\delta_{\rm sc}}(M,z) \to 1/2$. This would suggest that only half of the mass in the Universe ends up in collapsed objects. However, underdense regions can be enclosed within larger overdensities, giving them a finite probability of being within a larger collapsed object. Without a rigorous demonstration, Press and Schechter (1974) argued that initially underdense regions will eventually be accreted by collapsed objects and therefore added a "fudge factor" of 2 to their expression in Eq. (1.28). The *halo mass function*, defined as the number density of collapsed haloes as a function of mass, is then given by

$$\begin{aligned} \frac{\mathrm{d}n(M,z)}{\mathrm{d}M} &= 2\frac{\bar{\rho}}{M}\frac{\partial p_{>\delta_{\rm sc}}(M,z)}{\partial M} \\ &= \sqrt{\frac{2}{\pi}}\frac{\bar{\rho}}{M^2}\frac{\delta_{\rm sc}}{\sigma_M}\exp\left(-\frac{\delta_{\rm sc}^2}{2\sigma_M^2}\right)\left|\frac{\mathrm{d}\ln\sigma_M}{\mathrm{d}\ln M}\right|. \end{aligned} \tag{1.30}$$

An alternative, fully analytic derivation of the halo mass function in Eq. (1.30), which does not require inserting a fudge factor, was developed by Bond et al. (1991); it is known as the excursion set formalism, or Extended Press-Schechter (EPS) theory. Bond et al. (1991) argued that PS theory did not account for the 'cloud-in-cloud' effect i.e., smaller-scale underdensities living within larger-scale overdensities. A region that is locally underdense may still collapse into a halo, if it resides in a larger scale overdensity. A solution to this problem is found by evaluating the linear density contrast for a range of smoothing scales. This is known as a particle *trajectory* in the excursion set formalism. The fraction of collapsed haloes of mass M is equivalent to the fraction of trajectories with a *first upcrossing* of the density threshold barrier $\delta_{\rm sc}$ at mass scale M. Bond

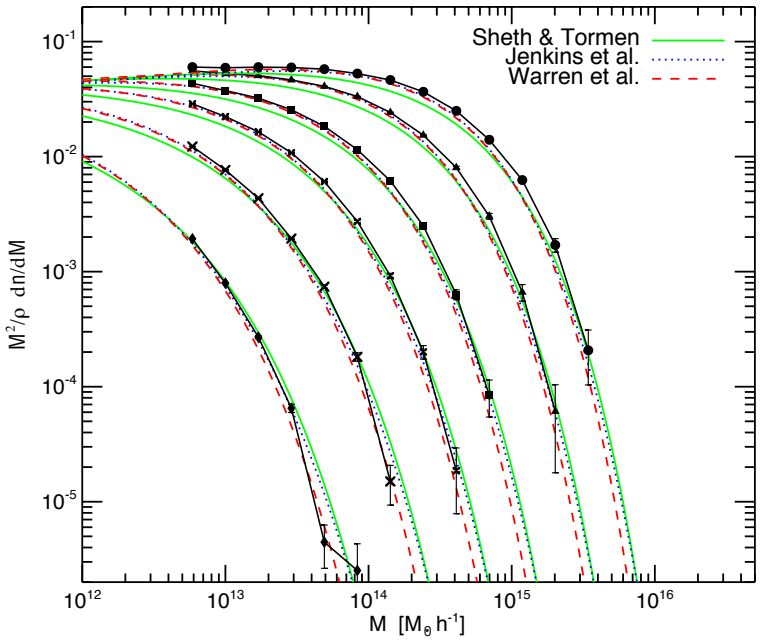

Figure 1.5: Comparison of Sheth and Tormen (1999), Jenkins et al. (2001) and Warren et al. (2006) halo mass functions with predictions from numerical simulations (black dots) at different redshifts. Figure taken from Grossi et al. (2009).

et al. (1991) derived an analytic expression of the halo mass function using a sharp k-space window function and recovered the PS halo mass function, including the factor of 2.

Subsequently, Sheth et al. (2001) extended Press-Schechter theory adopting arguments of ellipsoidal rather than spherical collapse. They wish to describe the evolution of an ellipsoidal perturbation in terms of three parameters: the initial ellipticity e, the prolateness p and the density contrast δ. In practice, they first compute the scale factor a at collapse as a function of e and p for a region with an initial overdensity $\delta = 0.04215$ in an Einstein-de Sitter universe. Since the linear theory growth factor is proportional to the scale factor in an Einstein-de Sitter model, this relation can be used to construct $\delta_{\rm ec}(e,p)$. They find that a reasonable approximation can by found by solving

$$\frac{\delta_{\rm ec}(e,p)}{\delta_{\rm sc}} = 1 + \beta \left[5\left(e^2 \pm p^2\right) \frac{\delta^2_{\rm ec}(e,p)}{\delta^2_{\rm sc}} \right]^{\gamma} \tag{1.31}$$

for $\delta_{\rm ec}(e,p)$. To further simplify Eq. (1.31), they assume the most probable values of e and p in a Gaussian random field, i.e. $p \approx 0$ and $e \approx (\sigma/\delta)/\sqrt{5}$ for regions with an initial value of δ/σ. As a result, they obtain an ellipsoidal collapse threshold which depends on the spherical mass variance

$\sigma(M)$ and is given by

$$\delta_{\rm ec} = \sqrt{a}\delta_{\rm sc}\left[1 + \beta\left(a\frac{\delta_{\rm sc}^2}{\sigma^2(M)}\right)^{-\gamma}\right] \tag{1.32}$$

where a, β and γ are free parameters calibrated to numerical simulations to yield an improved fit to the halo mass function. Similar to EPS, the fraction of collapsed objects of mass M is given by the fraction of density trajectories which upcross the ellipsoidal collapse threshold $\delta_{\rm ec}$ at scale $\sigma(M)$. This led to the establishment of the Sheth-Tormen model as the accepted theory of ellipsoidal collapse. In Chapters 3 & 4, we will test the interpretations of extended Press-Schechter and Sheth-Tormen theories using an independent approach to study halo formation based on machine learning.

More recently, different parametric functions for the halo mass function were derived to fit even better the halo abundance predicted by N-body simulations (e.g. Jenkins et al. 2001; Tinker et al. 2008; Warren et al. 2006). These parametric functions typically involve parameters which are calibrated with numerical simulations, which can vary in their volume and resolution or in their definition of haloes. Fig. 1.5 shows a comparison between Sheth and Tormen (1999), Jenkins et al. (2001) and Warren et al. (2006) halo mass functions and the simulation's predictions (black dots) at different redshifts.

1.4 Numerical simulations

Although linear perturbation theory can describe the growth of structures within certain limits, those calculations break down as density fluctuations grow large (i.e., $\delta > 1$) and cannot be used to explain much of the observational data from galaxy surveys. In fact, today's structures probe δ-values from a large dynamic range; from $\delta \sim -1$ in voids to $\delta \sim 10^7$ (or larger) in the densest regions of galaxies. Therefore, for a complete understanding of how the Universe evolved from tiny density fluctuations into stars, galaxies and galaxy clusters, one must also follow the evolution of the density field in the non-linear regime. Starting in the 1970s, it became possible to tackle this problem using numerical simulations, thanks to advances in computer performance and the development of sophisticated numerical algorithms. Today, numerical simulations play a major role in providing the bridge between theory and observation; the simulation gives us a prediction for the large-scale distribution of matter within the assumed cosmological model which can be tested against existing observations to assess the validity of the model. An example of a

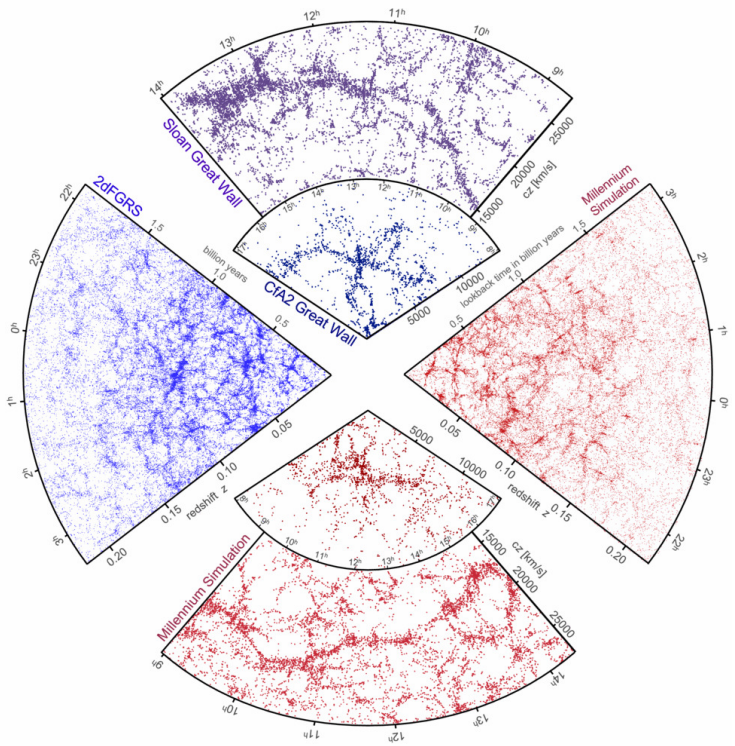

Figure 1.6: The galaxy distribution obtained from the SDSS and 2dFGRS spectroscopic redshift surveys (top and left panels) compared to mock galaxy distributions constructed using semi-analytic models within the dark matter distribution obtained from the Millennium simulation. Figure taken from (Springel et al. 2006).

comparison between the distribution of true galaxies observed from redshift surveys and mock galaxies constructed from the Millennium simulation is shown in Fig. 1.6; the success of numerical simulations at reproducing the observed large-scale structure is remarkable. The importance of numerical simulations in modern cosmology led to the establishment of its own new branch of cosmology, known as *computational cosmology*. In this section, I will give a general description of the N-body technique and its implementation by the `GADGET-3` code, used in the work presented in this thesis.

The goal of computational cosmology is to simulate the evolution of matter from early times, when density perturbations were small and well-approximated by linear theory, to the present epoch's non-linear regime. Most present-day simulations of the dark matter are obtained adopting the N-body approach, whereas baryonic matter and radiation require hydrodynamical simulations. In the ΛCDM model, the growth of structure is predominantly driven by the dark matter, interacting only via gravity. This allows us to study large-scale structure formation using simulations with only

dark matter particles, without the need to include the complicated (and more uncertain) physics of the baryon component. Although baryonic effects can have a significant impact on the dark matter distribution at small scales (e.g. Pontzen and Governato 2012), dark-matter-only simulations provide a good description of the real Universe on large cosmological scales.

A further aspect of simplicity in cosmological simulations is that the dark matter evolution can be very well described by Newtonian gravity, without the need for a full general relativistic treatment. The reason for this is straightforward at scales which are small compared to the comoving horizon size, as the effect of general relativity is expected to be negligible. However, on scales approaching the horizon size, GR provides the correct description of gravity and therefore one would expect clustering properties to be affected by various general relativistic effects. It turns out that for cold dark matter, additional terms in the equations of motion from GR cancel out. As a result, the Newtonian potential coincides with that of GR in the conformal Newtonian gauge even on very large scales (Chisari and Zaldarriaga 2011; Green and Wald 2012), similar to the case of linear order perturbation theory (see Sec. 1.3.1); Newtonian simulations can be safely used to study structure formation.

The basic principle of an N-body simulation is that the dark matter is traced by a set of macroscopic "particles" which interact with each other in a well-defined way (see e.g. Schneider 2006). The term particle refers to the smallest unit in the simulation, not to a fundamental particle in the particle physics sense; N-body particles are usually many orders of magnitudes heavier than fundamental particles, single stars, or even galaxies in the case of cosmological volumes. The particles are embedded in comoving cubes of length L, periodically extended to avoid particles at the edge of the box being influenced by the effect of the 'emptiness' outside the box. The basic ingredients of an N-body simulation are threefold; the size of the (comoving) volume $V = L^3$, the number of particles N and the cosmological parameters of the underlying cosmological model. The latter determine the initial conditions, the expansion history and the mass content of the virtual universe, whereas the box-size L and particle number N determine the range of accessible scales probed by the simulation.

The mass of each dark matter particle is given by

$$m_{\rm p} = \rho_{\rm crit}\Omega_m(L/N)^3, \tag{1.33}$$

where $\rho_{\rm crit} \sim 27.8 \times 10^{10}\, h^{-2}\, {\rm M}_\odot\, {\rm Mpc}^{-3}$. An increased mass resolution is achievable by increasing the number of particles, which significantly increases the computational cost, and/or by reducing

the volume, resulting in fewer objects especially at large scales. The key is to find a balance between computational power and numerical accuracy to resolve the scales of interest.

A collisionless system, such as that of cold dark matter, the equation of motion is dictated by the total gravitational force on a particle i given by

$$sF_i = \sum_{i \neq j} \frac{GM_{\rm p}^2(r_j - r_i)}{\left(|r_j - r_i|^2 + \epsilon^2\right)^{3/2}}, \tag{1.34}$$

where the j-th particles are all other particles in the simulation such that $|r_j - r_i| > \epsilon$, where ϵ is an arbitrary small number known as the *softening length* (Dehnen 2001). The softening length is introduced to modify the gravitational force at small scales in order for strong particle-particle collisions not to affect the dynamics of the system. It is usually taken to be the mean separation between two particles in the box. At scales below the softening scale, the gravitational force is modified and the simulation cannot be trusted; it therefore sets a spatial resolution limit for the simulation. The major shortcoming of solving Eq. (1.34) is computational time, which scales as N^2 with N representing the number of particles. This makes such 'brute force' method unsatisfactory for cosmology, where we aim to simulate the evolution of $N > 10^6$ particles. In Chapters 3, 4 and 5, we make use of N-body simulations based on the GADGET-3 code (Springel 2005); I will therefore focus on reviewing its implementation of the gravitational forces.

1.4.1 The `GADGET` code

Sufficiently accurate algorithms have been developed to make use of approximations, allowing for much faster evaluation of the gravitational forces. The *particle-mesh method* (Hockney and Eastwood 1988) and the *Barnes-Hut tree algorithm* (Barnes and Hut 1986) are the two most widely used. `GADGET-3` uses a TreePM code, a hybrid algorithm employing different ways of calculating the forces between particles on large and small scales (Springel 2005; Springel et al. 2001). On large scales, the force is calculated via the fast Particle-Mesh (PM) method, and on small scales via the slow but precise Barnes-Hut (BH) tree method.

The PM method obtains the evolution of the system by solving the Poisson equation,

$$\nabla^2\Phi(\boldsymbol{x}, t) = 4\pi G a^2(t)\left[\rho(\boldsymbol{x}, t) - \overline{\rho}(t)\right]. \tag{1.35}$$

The first step is to compute the density field ρ by placing particles onto a uniform grid, or "mesh". It then employs fast Fourier transforms (FFT) to speed up the solution for the potential Φ in Poisson's

equation, reducing the computational cost to $\mathcal{O}(N_{\rm grid} \log N_{\rm grid})$. The force is then obtained by computing the gradient of the potential and it is applied to each particle using the same initial grid. For small particle separations, the numerical accuracy of the PM method quickly degrades. Note that the Fourier approach implicitly assumes periodic boundary conditions for the computational domain, exactly as desired for cosmological simulations of structure formation.

The BH tree code is instead used to solve the equations on small scales. It reduces to direct summation for the contribution of nearby particles, but involves a great simplification in the evaluation of long-range interaction. It uses a hierarchical tree algorithm that groups particles into cells called tree nodes. Each node is in turn sub-divided into further sub-nodes (or, *leaves*), until the lowest level of the tree is reached, where each leaf contains one or zero particles. In this way, the force acting on a particle is calculated by summing the partial forces from neighbouring tree nodes only, instead of requiring $N - 1$ partial forces per particle as for the direct-summation approach. Once the forces are computed, the equations of motion are numerically integrated over time using a leap-frog integration scheme.

1.4.2 Initial conditions

The purpose of initial conditions is to provide a discrete representation of the primordial density field predicted by the cosmological model, to serve as the starting point of the actual cosmological simulation. In the simplest inflationary models, the initial density fluctuations can be characterized by a Gaussian random field, meaning that only the power spectrum $P(k)$ is required to provide a complete description of its statistical properties. As discussed in Sec. 1.2, this has been observationally confirmed to a high level of precision by measurements of the CMB. The initial density field of a simulation is therefore given by a realisation of a Gaussian random field with a power spectrum $P(k)$ scaled by the primordial power spectrum and the transfer function as

$$P(k) = \alpha k^{n_s} T^2(k), \tag{1.36}$$

where n_s is the spectral index of the primordial power spectrum, α is a normalisation constant and $T(k)$ is the transfer function at the starting redshift, typically computed numerically by Boltzmann solvers such as CAMB (Lewis et al. 2000). The initial conditions are usually set at a high redshift (typically, $z \sim 100$) such that the fluctuations are still in the linear regime but also well inside the matter dominated era.

In practice, generating the initial conditions involves two main steps. First, the N-body particles

are distributed on a uniform grid. The particles are then moved slightly away from the grid positions to create density perturbations. This can be done using the Zel'dovich approximation (Zel'dovich 1970),which relates a particle's Eulerian position $\mathbf{x}$ to its Lagrangian position $\mathbf{q}$ via

$$\mathbf{x}(\mathbf{q}, t) = \mathbf{q} + D(t)\mathbf{f}(\mathbf{q}), \tag{1.37}$$

where the displacement field $\mathbf{f}(\mathbf{q})$'s Fourier modes are given by

$$\mathbf{f}_\mathbf{k} = -i\frac{\delta_k}{k^2}\mathbf{k}. \tag{1.38}$$

As a result, the initial conditions assign an initial position and velocity to each particle given the initial power spectrum and the cosmological background.

1.4.3 Finding dark matter haloes

At the end of the simulation one is left with a prediction for the large-scale distribution of matter within the assumed cosmological model; this could in principle be tested against existing observations in order to assess the validity of the model or to constrain its parameters. It is therefore useful to develop algorithms which identify dark matter haloes given the matter distribution in the simulation.

Historically, there have been two major approaches to identify haloes in numerical simulations: Spherical Overdensity (SO) and Friends-of-Friends (FoF). The first can be categorized as a density peak locator, whereas the second as a particle collector. Density peak locators identify haloes via a two step approach: (i) peaks are identified in the matter density field and (ii) spherical shells about these centres are grown out until the density profile drops below a certain value. The latter is usually derived from the spherical collapse model. Most of the methods utilising this approach differ in the way they locate density peaks. Particle collectors instead connect and link particles together that are closer than a given threshold (either in a 3D configuration or in 6D phase-space). Both methods usually end with a step where gravitationally unbound particles are removed from the halo. In Chapters 3, 4 and 5, we make use of `SUBFIND` (Springel 2005), a FOF algorithm that also identifies substructures within the parent halo, and the `Amiga Halo Finder` (AHF, Gill et al. 2004; Knollmann and Knebe 2009), which employs a recursively refined grid to locate local overdensities in the density field. The identified density peaks are then treated as centres of prospective haloes and sub-haloes.

Fundamental halo properties such as the halo mass are found to agree remarkably well across a wide range of dark matter halo finders. Recent years have seen increased activity in developing new algorithms, often using the full phase-space information of the particle distribution, or other more sophisticated techniques. The advantages of phase-space based techniques only become apparent when looking for example at substructures (Avila et al. 2014) and in reconstructing merger trees (Muldrew et al. 2011). For a detailed comparison of different halo finding algorithms, see Knebe et al. (2011, 2013).

1.5 Outline of the thesis

This thesis is organized as follows. In Chapter 2, I describe the necessary background to the machine learning methods used in this thesis. I will introduce ensemble methods, specifically random forests and gradient boosted trees, and convolutional neural networks. Chapters 3 to 5 describe the developments of a machine learning approach to gain new physical insights into dark matter halo formation. The approach consists of training a machine learning algorithm to learn the relationship between the initial conditions and the final dark matter haloes from N-body simulations. In Chapter 3, I will present the first application of our method; halo formation is turned into a binary classification framework, where the machine learning algorithm classifies dark matter particles in an N-body simulation into two classes, depending on whether or not they will form haloes above a specified mass threshold at $z = 0$. I will show how this leads to a different interpretation of the role of the tidal shear field in halo collapse, which differs from existing interpretations based on analytic approximations. In Chapter 4, the framework is generalized to a regression problem, in order to investigate haloes across a wider range of final mass. In Chapter 5, I will describe ongoing work on developing a framework based on convolutional neural networks, able to automatically extract features relevant to halo formation from the initial conditions density field. By developing tools that allow for the interpretability of the results from convolutional neural networks, we hope to uncover new physical relations between the initial conditions and the final dark matter haloes. The latter is part of future work, discussed in Chapter 6, which focuses on the idea of *knowledge extraction* in machine learning applied to cosmological structure formation; we plan to extract information from the deep learning model regarding the underlying physics driving the formation of large-scale structures. In Chapter 7, I will outline the conclusions of this thesis.

This thesis contains material from the following two papers, together with ongoing work:

- *Machine learning cosmological structure formation.* This work was published as Luisa Lucie-

Smith, Hiranya V. Peiris, Andrew Pontzen, and Michelle Lochner, Monthly Notices of the Royal Astronomical Society, Volume 479, Issue 3, September 2018, Pages 3405–34146 and was carried out in collaboration with the named co-authors.

- *An interpretable machine learning framework for dark matter halo formation.* This work was published as Luisa Lucie-Smith, Hiranya V. Peiris, and Andrew Pontzen, Monthly Notices of the Royal Astronomical Society, Volume 490, Issue 1, November 2019, Pages 331–342, and was carried out in collaboration with the named co-authors.

2 Method

2.1 Machine learning algorithms

The basic idea behind machine learning algorithms is to identify the relationship between the input and output data of some set of samples, known as the *training set*[1]. Machine learning is usually employed for problems that are so highly non-linear that they can not be solved using traditional (analytic or numerical) statistical techniques. The advantage of machine learning is that it can automatically build a model up to an arbitrary level of complexity. Provided the training set is a representative sub-sample of the data, the trained model can then be used to make predictions on new unseen data. Typically, it is difficult for the algorithm to learn patterns in the training set if the input data is made of noisy and high-dimensional data. Particularly now in the era of "Big Data", we are faced with large amounts of high-dimensional data. Feature extraction and feature selection are two powerful tools which can address these issues by projecting the original high-dimensional space into a new low-dimensional feature space (feature extraction) and by selecting only a subset of informative features to construct the algorithm (feature selection) (Li et al. 2016). Both tools can provide significant improvements in the algorithm's performance[2], as well as requiring lower computational and memory costs.

The two main categories of machine learning techniques are *supervised learning*, where the training set is given by a set of input features and their corresponding label, and *unsupervised learning* where the training set consists of features with no corresponding label. Typically supervised learning is used for classification problems, where the training samples belong to two or more

[1]For good introductions to the field of machine learning see Abu-Mostafa et al. (2012); Bishop (2006); Hastie et al. (2005); MacKay (2003); Murphy (2012).

[2]Note that whilst feature extraction is indispensable for all machine learning algorithms (with the exception of convolutional neural networks), not all require feature selection; some are robust to a large number of uninformative features.

classes, or regression problems, where the output consists of continuous variables. Unsupervised learning is usually used in clustering problems, where the aim is to group similar samples in the data. In addition, *semi-supervised* learning algorithms, which involve a mixture of labelled and unlabelled training samples, can also be used in problems such as image recognition. I will focus on supervised learning algorithms, as these are most relevant for this thesis.

All machine learning models are affected by a trade-off in their ability to minimize bias and variance in the predictions (Geman et al. 1992; Hastie et al. 2005). The bias is the difference between the average prediction of the model and the correct value we are trying to predict, whereas the variance is the variability of the model's prediction for a given correct value (or sample). A model that under-fits the data generally has a high bias and a low variance, whereas one that over-fits has a low bias and a high variance. The key is to find the right balance in the complexity of the model which neither over-fits nor under-fits the data; this trade-off in complexity gives rise to the trade-off between bias and variance. An understanding of these errors can help with choosing the appropriate machine learning algorithm for a given problem and to build accurate models.

Early applications of machine learning to astronomy were primarily focused on observational tasks. One of the first applications involved using neural networks to distinguish galaxies from stars in photometric catalogues (Bertin 1994; Maehoenen and Hakala 1995; Odewahn et al. 1992). This method was later incorporated in `SExtractor`, a widely-used software that classifies sources from astronomical images (Bertin and Arnouts 1996). Machine learning tools were also found successful in classifying both stellar spectra (von Hippel et al. 1994) and galaxy spectra (Folkes et al. 1996). Artificial neural networks were used for the first time as an automated tool for the morphological classification of galaxies, able to reproduce visual classifications of galaxies made by humans (Banerji et al. 2010; Lahav et al. 1995; Lahav et al. 1996; Naim et al. 1995; Storrie-Lombardi et al. 1992). Subsequently, machine learning gained prominence as a successful tool for calculating photometric redshift using a variety of models, including neural networks (Ball et al. 2004; Firth et al. 2003), which led to the development of the well-known `ANNz` code for photometric redshift estimation (Collister and Lahav 2004), as well as support vector machines (Wadadekar 2005), decision trees (Carliles et al. 2008) and k-nearest neighbours (Ball et al. 2008). More recently, machine learning has proved to be a successful tool for a much broader range of applications in cosmology and astrophysics, and beyond (see e.g. Ball and Brunner (2010); Mehta et al. (2019) for reviews of machine learning in astronomy). These include a variety of tasks for large-scale structure analyses (Berger and Stein 2019; Charnock et al. 2019; He et al. 2019; Kodi Ramanah et al. 2019; Mathuriya et al. 2018; Merten et al. 2019; Modi et al. 2018; Ntampaka et al. 2019; Pan et al. 2019;

Ravanbakhsh et al. 2016; Zhang et al. 2019), fast automated object identification (Lochner et al. 2016; Moss 2018), gravitational lensing studies (Gupta et al. 2018; Jeffrey et al. 2019; Peel et al. 2019; Schmelzle et al. 2017), gravitational waves (Dreissigacker et al. 2019; Gebhard et al. 2019) and mass estimates of galaxy clusters (Ntampaka et al. 2015) and the Local Group (McLeod et al. 2017)[3]. Deep learning is the fastest growing branch of machine learning. Recent advances have led to models that reach human performance across diverse areas of science, industry and academia, such as image, sound and text recognition, as well as robotics and game play tasks (Silver et al. 2016). In the context of cosmological simulations, Ravanbakhsh et al. (2016) were the first to apply deep learning techniques to estimate cosmological parameters from the 3D dark matter distribution in an N-body simulation. Their work was then extended to different deep learning architectures (e.g. Mathuriya et al. 2018; Pan et al. 2019) and to estimate cosmological parameters from 3D simulated galaxy maps (Ntampaka et al. 2019). Other applications of deep learning to cosmological simulations typically involve learning fast mappings which would otherwise require expensive N-body simulations, such as the mappings between the Zel'dovich-displaced and the non-linear density fields (He et al. 2019), the non-linear density field and the halo distribution (Charnock et al. 2019; Kodi Ramanah et al. 2019; Modi et al. 2018) and the dark matter and galaxy distributions (Zhang et al. 2019).

However, understanding the inner workings of machine learning models, especially in the context of deep learning, remains a challenge. Often, the best performing models are so complex that they lack of transparency and are hence considered to be "black-boxes". On the other hand, simpler algorithms such as linear regression models are straightforward to interpret as they are based on simple and smooth relationships between the inputs and outputs, but are limited in complexity. This leads to an accuracy vs. interpretability trade-off in machine learning (Kuhn and Johnson 2013); black-box models provide great accuracy but make it hard to understand what information the algorithm detects, or how the algorithm produces its outputs. Moreover, the features interact in such a complex and highly non-linear manner, that it is difficult to provide an estimate of the importance of individual features for the algorithm's learning. In science, we require the ability to explain how and why certain predictions are made by a model, thus making the field of interpretability of machine learning algorithms of primary importance. Developing techniques to turn "black-box" algorithms into interpretable ones is essential for machine learning applications to cosmology; ultimately, it will allow us to interpret machine learning results in terms of the underlying physics.

[3]For a recent review on future prospects of machine learning in cosmology see Ntampaka et al. (2019).

One special family of machine learning algorithms that provide excellent accuracy with very high interpretability are ensemble methods. In this section, I will first review random forests and gradient boosted trees, the ensemble methods used in Chapter 3 & 4, and then move on to describe deep convolutional neural networks (CNNs), used in Chapter 5 & 6, for which interpretability is a challenge.

2.1.1 Decision trees

A decision tree is a supervised learning method which predicts the output of a sample by following a set of simple decision rules inferred from the features of the data, **X** (Breiman et al. 1984; Hastie et al. 2005; Quinlan 1986; Salzberg 1994). A tree is formed by a set of nodes, each with its own decision rule of the form $X_i \leq n$, where X_i is the feature which makes the split and n is some value of that feature that determines the split between the samples. An illustration of a (shallow) decision tree is shown in Fig. 2.1. During inference, samples at a node will be split into left and right nodes according to their value for the feature X_i. This process is repeated for each node of the tree until one reaches a *leaf node*, where no more splits are made. The final leaf node returns the final prediction for all samples that end up in such leaf. In a classification task, this is the probability of belonging to each class which comes from the fraction of training samples in each class at that leaf node. In regression, the final prediction is a single value of the (continuous) target variable given by the average value from the training samples that end up in that leaf node.

Training a decision tree is equivalent to selecting decision rules which optimally partition the training data. This requires adopting a metric to define the best split. Different metrics exist to choose the best feature and the best value for that feature at any given node of the tree. We follow the implementation in `scikit-learn` (Pedregosa et al. 2011), where the best decision rule is one that maximizes the decrease in *impurity*, a measure of the error in the predictions. Mathematically, this is defined as follows. Consider a node n of a decision tree, where the split made by feature X divides N_n samples into two subsets of N_{n_L} and N_{n_L} samples in the children nodes n_R and n_L, respectively. The splitting feature X is chosen such that it maximizes the impurity decrease $\Delta p(n)$, defined as

$$\Delta p(n) = p(n) - \frac{N_{n_L}}{N_n} p(n_L) - \frac{N_{n_R}}{N_n} p(n_R), \tag{2.1}$$

where $p(n)$, $p(n_L)$ and $p(n_R)$ are the impurity measures at node n, n_L and n_R, respectively. For a

classification problem, the impurity p can be the Gini index,

$$p_G(n) = 1 - \sum_{j=1}^{c} s_j(n)^2, \tag{2.2}$$

or the Shannon entropy (or, information gain),

$$p_E(n) = -\sum_{j=1}^{c} s_j(n) \log_2 s_j(n), \tag{2.3}$$

where $s_j(n)$ is the proportion of samples that belong to class j at node n and c is the total number of classes. Both the Gini index and entropy are zero when the node is pure i.e., when all samples belong to one class, and are maximum when the samples are evenly distributed in classes. For regression, the impurity p is usually given by the mean squared error or the mean absolute error.

All machine learning models, including decision trees, have certain parameters that are not tuned during the algorithm's learning process but must be manually set a priori. These are called *hyperparameters* and need to be optimized for any given training set. Generally, these are parameters that describe high-level characteristics of the model. Examples of hyperparameters in decision trees are the maximum depth of the tree, the impurity measure at each node (entropy or Gini index) and the minimum number of samples at a leaf node. To optimize the hyperparameters, one possible strategy is to construct grids of values for each hyperparameter, consider all possible combinations of parameters and choose the setting which performs best on an independent set of samples, known as the *validation set*, according to some evaluation metric. The fact that the choice of hyperparameters is tested on an independent validation, rather than on the training set, avoids overfitting to the latter. However, the problem with this optimisation process is that one risks overfitting to the validation set; it possible that the selected hyperparameters are the optimal fit to the validation set, but do not generalize to independent dataset.

One common approach to minimize overfitting is known as k-fold *cross-validation* (James et al. 2014; Kohavi 1995; Rao and Fung 2008). Here, each combination of hyperparameters is tested on a number of validation sets taken, rather than on just one. The training set is divided into k smaller sets. For each k-fold, k-1 sets are used for training and one is used as a validation set which measures its performance. This procedure is repeated k times so that each set is used as a validation set once, where typical values of k are 5 and 10. The hyperparameter combination that retains the best score on the validation test is then kept by the estimator. One can then check if this combination is overfitting or not by comparing the performance score of the validation test

with that of the test set left out of the training process. If the validation set score is larger than the test set score, then the classifier is overfitting. Otherwise, that setting is the optimal choice for that given training set.

Given a set of values for the hyperparameters, the training set is divided into k equal-sized sets where $k-1$ sets are used for training and one is used as the validation set. The training/validation procedure is repeated k times such that each time a different k set is kept aside for validation and the rest are used for training. Finally, the score for that set of hyperparameters is given by the average score over the k validation sets. The hyperparameter combination that retains the highest average score on the validation sets is then kept by the estimator. In summary, the main benefits of k-fold cross validation are twofold. First, setting aside a subset of the training set for validation ensures that the hyperparameters of the algorithm do not overfit the training data. Second, averaging the score over k validation sets also ensures that the hyperparameters do not overfit any single validation set. Variations of this method for hyperparameter tuning also exist, as for example randomized grid-search (Bergstra and Bengio 2012) or other model-specific techniques.

Decision trees are often referred to as a "white-box" model, as it is simple to understand their inner working and and to interpret their predictions. This simplicity comes at the expense of accuracy; they typically create over-complex trees which provide an excellent fit to the training data, but do not generalize to new, independent data. Mechanisms such as pruning, setting the minimum number of samples required at a leaf node or setting the maximum depth of the tree can partially help to mitigate this problem but are often not enough to fully avoid it. In addition to this, small variations in the training data lead to completely different tree being generated, leading to a model with high variance error in the predictions.

A better solution comes from combining a large number of individual decision trees into ensemble learners. In the following sections, I will introduce ensembles of decision trees and two specific types of ensembles used in our work; *random forests* and *gradient boosted trees*.

2.1.2 Ensembles of decision trees

Since individual trees generally over-fit the training data, they are often combined together to form a more robust ensemble estimator. The two main approaches to combine decision trees are *bagging* (Breiman 1996) and *boosting* (Freund and Schapire 1996). The two approaches form ensembles that differ substantially in the trade-off between the models' ability to minimize bias and variance in the predictions. Bagging estimators are effective at decreasing variance, but have no effect on the

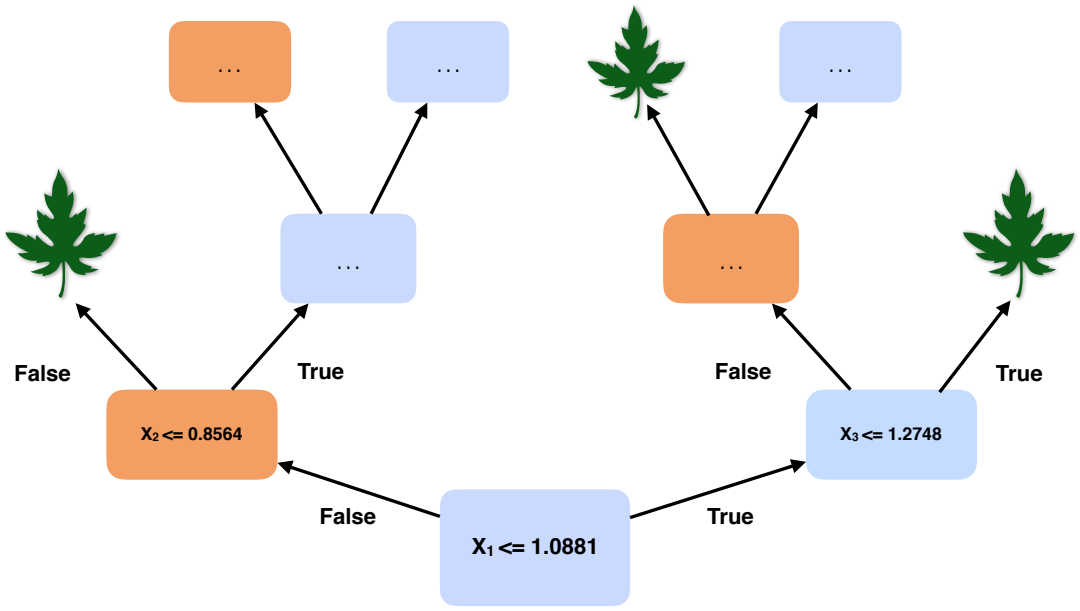

Figure 2.1: An illustration of a decision tree, with nodes filled by decision rules inferred from the features of the training data. Inference is made on unseen samples by following the decision rules until they reach a leaf node, where no more splits are being made and the algorithm makes its prediction.

bias; trees learn independently on bootstrapped training samples and the final prediction of the ensemble is given by the average over individual trees' predictions. On the other hand, boosting can reduce both the bias and the variance contributions to the error in the predictions (Schapire et al. 1998) by aggregating trees iteratively, in such a way that subsequent trees learn to correct the mistakes of the previous ones.

Random Forests

Random forests are an ensemble of decision trees which combine trees by means of bagging (Breiman 2001). They are found to outperform most other popular machine learning algorithms (as for example Naive Bayes, Support Vector Machines and k-nearest neighbours) for many classification problems (Caruana and Niculescu-Mizil 2006). We made use of the Python `scikit-learn` (Pedregosa et al. 2011) package's implementation of random forests.

Ensemble methods usually work by combining predictions of several base estimators in order to improve generalisability over a single estimator (Dietterich 2000a,b). Random forests are formed by an ensemble of decision trees, each acting in parallel on the data and equally contributing to the final prediction. The prediction of the ensemble is given by the average of the predictions from

individual trees. In the case of classification, this is the probability of belonging to each individual class, whereas for regression it is a single value for the target variable.

The power of random forests to reduce variance only manifests when randomness is introduced in order to reduce correlations between the classifiers within the ensemble. The main elements of randomness in a random forest are twofold. The first is that each tree of the forest is trained on a random subset of samples drawn with replacement from the original training set. This is a common feature amongst all bagging estimators. The second is that the best feature at a node of a single tree is chosen not out of all features, but out of a sub-set of randomly drawn features. Using feature bagging reduces correlations between decision trees that can arise when only a few features are strongly predictive of the final output. A third element of randomness can be introduced by splitting features randomly rather than using the criteria described above; these are known as extremely randomized forests. Although individual trees become even weaker estimators, the ensemble predictive power can be increased as it reduces correlations between trees even further. In general, a random forest can slightly increase the bias (with respect to the bias of a single non-random tree) but its variance dramatically decreases due to averaging over the trees; the latter usually more than compensates for the increase in bias, hence yielding an overall better model.

Similar to decision trees, random forests also have hyperparameters that need to be optimized using cross-validation methods. These include all hyperparameters of decision trees, as well as the number of trees in the forest and the fraction of features to randomly draw at each node when looking for the best split.

Gradient boosted trees

Gradient boosted trees are a boosting ensemble of decision trees (Freund and Schapire 1997; Friedman 2001, 2002). The main difference with random forests is that trees are added one at a time to the ensemble, such that each new tree acts to correct errors made by the existing ensemble. This is in contrast to bagging, where the contribution of all predictors is weighted equally in the bagged ensemble. The basic idea of gradient boosted trees is to combine the idea of boosting and gradient descent optimization to construct the ensemble. The performance of gradient boosted trees can be expressed in terms of the loss function; the aim of the algorithm is to minimize the loss evaluated for the training data by adopting a gradient-descent optimization procedure. At each step, the algorithm computes the gradient of the loss function with respect to the predicted value of the ensemble and adds trees that move the loss in the direction of the gradient. In practice, this

requires a clever way of mapping gradients to decision trees.

Mathematically, this can be described as follows. At any given iteration m in the gradient boosted tree, a new decision tree $f_m(\mathbf{x})$ is added to the existing ensemble $F_{m-1}(\mathbf{x})$ such that the prediction for a given training sample i, $F_m(\mathbf{x}_i)$, is updated as

$$F_m(\mathbf{x}_i) = F_{m-1}(\mathbf{x}_i) + f_m(\mathbf{x}_i), \tag{2.4}$$

where $\mathbf{x}_i$ is the input vector for that training sample. An illustration of the iterative addition of decision tree learners in order to construct a gradient boosted tree is shown in Fig. 2.2. The accuracy of the gradient boosted tree is quantified by the loss function, measuring how well the model's learnt parameters fit the data. The aim is to build a sequence of M trees which minimizes the loss function between the target value y and the predicted one $\hat{y} = F_M(\mathbf{x})$. Gradient boosted trees solve this minimization problem using gradient-descent optimization. The parameters of a decision tree, consisting of both the decision rules and the target variable for that tree, are chosen to point in the direction of the negative gradient of the loss function with respect to the ensemble's predictions. As an example, consider a regression task with the loss function to be the mean squared error between the target value y and the prediction $\hat{y}$. At iteration m, the loss function L is given by the mean squared error between the target value y and the current prediction $\hat{y} = F_{m-1}(\mathbf{x})$ for N training samples,

$$L(y, F_{m-1}) = \sum_{i}^{N} \frac{\left(y_i - F_{m-1}(\mathbf{x}_i)\right)^2}{2}. \tag{2.5}$$

The negative gradient of the loss function with respect to the predictive model for each training sample i is given by

$$r_i = -\frac{\partial L(y, F_{m-1})}{\partial F_{m-1}}\bigg|_i = y_i - F_{m-1}\left(\mathbf{x}_i\right). \tag{2.6}$$

Therefore, when choosing the mean squared error as the loss function, the decision tree at iteration m is trained to predict the residuals r of the current predictions with respect to the true target values. This procedure is repeated until adding further trees does not yield further changes in the loss. Gradient boosted trees are flexible enough to minimize any loss function, as long as it is differentiable.

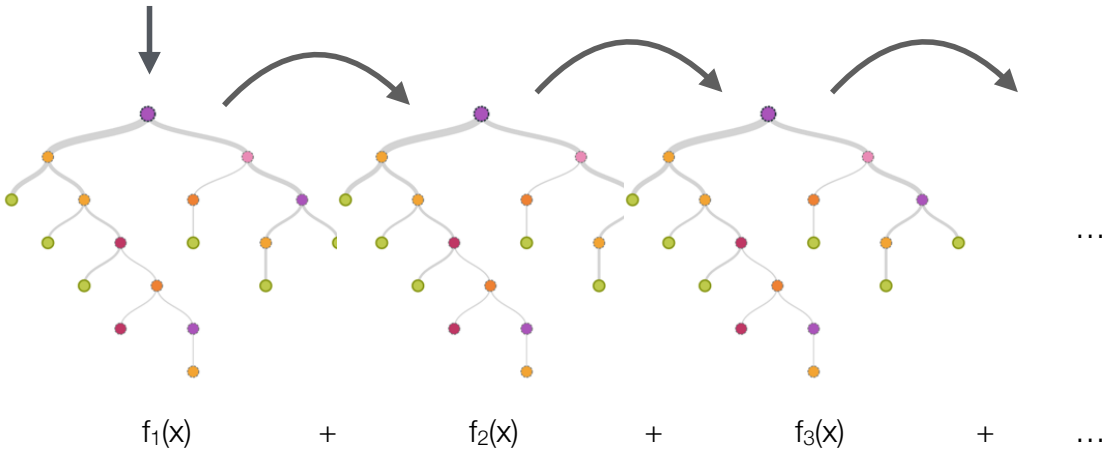

Figure 2.2: An illustration of a gradient boosted tree, where new decision trees are iteratively added to the existing ensemble following a gradient-descent optimization procedure. Re-adapted from `https://bigml.com`.

2.1.3 Feature importances

Ensembles of decision trees provide an excellent trade-off between interpretability and accuracy. In addition to the predictive power of ensembles of trees, they also allow for interpretability of their learning procedure. This can be achieved using a metric known as *feature importances* (Louppe et al. 2013) to measure the relevance of each input feature in training the algorithm to predict the correct output. This is a crucial aspect of our machine learning application; it will allow us to extract physical knowledge from the machine learning results.

The importance of the j-th feature X_j from a single tree t of the ensemble is given by

$$\mathrm{Imp}_t\left(X_j\right) = \sum_{n \in \{n \text{ is split on feature } X_j\}} \frac{N_n}{N_t}\left[p - \frac{N_{n_R}}{N_n} p_R - \frac{N_{n_L}}{N_n} p_L\right], \tag{2.7}$$

where N_t, N_n, N_{n_R}, N_{n_L} are the total number of samples in the tree t, at the node n, at the right-child node n_R and at the left-child node n_L, respectively. The sum in the equation is over all n nodes where the feature X_j makes the split. The impurity p is given by the choice of splitting criterion, which in our case is the mean squared error. The final importance of feature X_j given by

the ensemble of T trees is the normalized sum over the importances from all trees,

$$\text{Imp}(X_j) = \frac{\sum_{t=1}^{T} \text{Imp}_t(X_j)}{\sum_{j=1}^{J} \text{Imp}(X_j)}. \tag{2.8}$$

2.2 Deep learning algorithms

Neural networks are a machine learning model inspired by the way biological neural networks process information in the human brain (Nielsen 2015). Deep neural networks have a long history (Bishop 1995), but the notion of *deep learning* was introduced in the 2000s when large artificial neural networks were used for Boltzmann machines (Hinton et al. 2006; Salakhutdinov and Hinton 2009). The word *deep* refers to the hierarchical structure in neural networks, where many stacks (or, layers) of neurons are placed between the inputs and the outputs (Bengio 2009; Goodfellow et al. 2016). The outputs of the first layer become the inputs of the second layer, and so on through each layer in the network until the final output (Deng and Yu 2014). The output from each layer is distinctly different and more abstract than the original input data; this level of increasing abstraction introduced by each layer makes deep learning algorithms more difficult to understand than standard machine learning algorithms.

Deep learning networks quickly demonstrated their success compared to shallow machine learning algorithms. The first examples of this came from two deep learning implementations, the first being AlexNet (Krizhevsky et al. 2012), which reduced the error on the ImageNet Large Scale Visual Recognition Challenge by 12%, and the second ResNet, which also achieved dramatic improvement over existing methods with an error of 3.57% (He et al. 2015). Since then, deep neural networks have become the standard technique of many image and speech recognition tasks (see e.g. Mehta et al. (2019) for a review). Neural networks also form the backbone of CNNs (Lecun and Bengio 1995), which are the main focus of Chaper 5. We first give an overview of neural networks and then describe in detail the workings of CNNs.

2.2.1 Neural networks

The fundamental unit of a feed-forward neural network is a neuron, which takes scalar inputs, performs first a linear transformation and then a non-linear one, and finally outputs a real number. The linear transformation takes the form of a dot product with a set of neuron-specific weights and

an additional bias, as

$$z_i = \mathbf{w}_i^{\mathrm{T}} \cdot \mathbf{x} + b_i, \tag{2.9}$$

where z_i is the output, $\mathbf{x}$ is the input, $\mathbf{w}_i$ are the weights of the i-th neuron and b_i is the bias. A non-linear transformation is then applied to z_i, of the form $a_i = \sigma(z_i)$, where σ is given by some choice of non-linear function. Historically, common choices of non-linearities included step-functions (perceptrons), sigmoids and the hyperbolic tangent. More recently, it has become more common to use rectified linear units (ReLUs), leaky rectified linear units (leaky ReLUs), and exponential linear units (ELUs), as they are found to outperform other choices in a variety of tasks (Nwankpa et al. 2018). The output $a_i(\mathbf{x})$ serves as input to the next layer, and so on, until one reaches the output layer. Typically, a neural network is organized by stacking a number of "hidden" layers in between the input and output layers, thus forming a *deep* neural network as shown in Fig. 2.3. Each hidden layer is in turn made up of a large number of neurons; the power of neural networks stems from how neurons are connected to each other. Since all neurons of neighbouring layers are connected to each other, the layers are also known fully-connected layers.

In summary, the deep neural network can be thought of as a complicated non-linear transformation of the inputs $\mathbf{x}$ into an output y that depends on the weights and biases of all the neurons in the input, hidden, and output layers. The predictive power of neural networks expresses itself in the universal approximation theorem, which states that a neural network with a single hidden layer can approximate any continuous, multi-input/multi-output function with arbitrary accuracy (Nielsen 2015). However, the more complicated a function, the more hidden units (and free parameters) are needed to approximate it (at fixed accuracy). Hence, the applicability of the approximation theorem to practical situations is limited by computational power, training data availability and other factors.

The basic principles of training a neural network are somewhat similar to those of gradient boosted trees; one specifies a loss function and then updates the parameters, in this case the weights and biases, using gradient descent optimization. First, information is propagated through the network in a feed-forward fashion i.e., forward layer-by-layer through the network from the inputs to the outputs. Unlike gradient boosted trees, the optimization of the parameters requires a more complicated procedure, known as *backpropagation* (Rumelhart et al. 1986), due to the large number of weights and biases in the different hidden layers and the complexity of deep neural networks. At its core, backpropagation is simply the ordinary chain rule for partial differentiation applied to solve the gradient of the loss with respect to the weights and biases. The term backpropagation comes

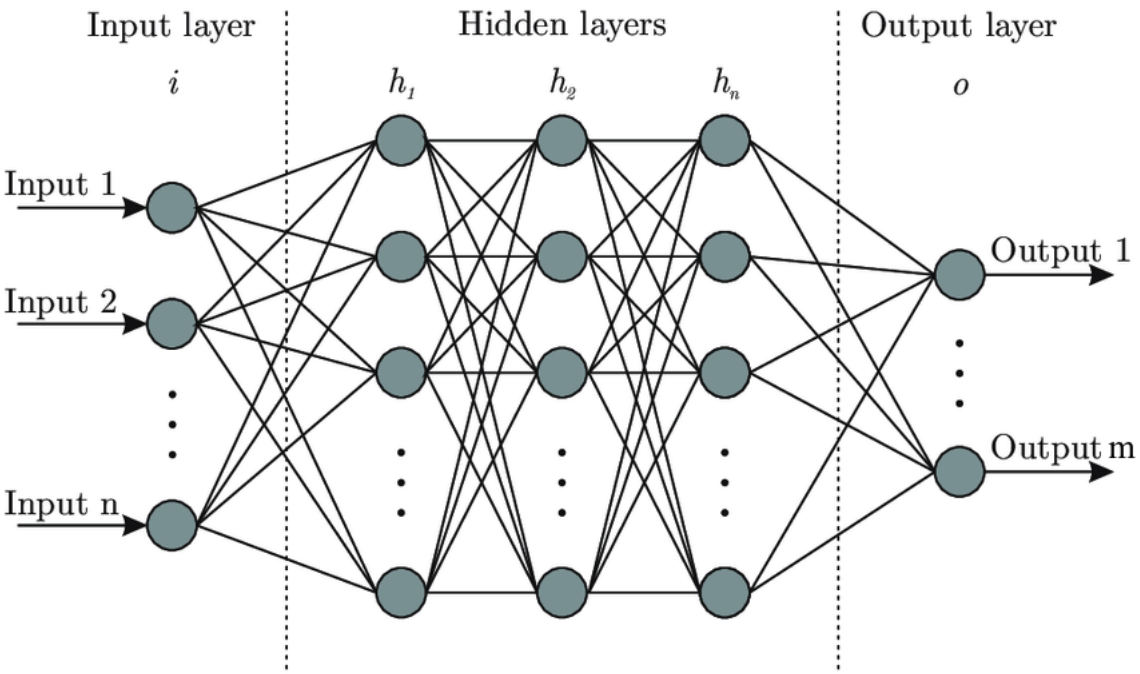

Figure 2.3: An illustration of a deep neural network with an input layer, three hidden layers and an output layer. Figure re-adapted from `https://www.wandb.com/articles/fundamentals-of-neural-networks`.

from the fact that in computing the gradients one moves back from the outputs to each hidden layer, until reaching the input layer. Forward and backward passes of the neural network is typically done many times, where each pass is known as an *epoch*. At each epoch, new updates to the weights are being made in the direction of the negative gradient of the loss with respect to the weights. The convergence of the neural network can be tested by tracking the loss function as a function of number of epochs; once the loss does not change with increasing number of epochs, then the algorithm has reached convergence.

2.2.2 Deep convolutional neural networks

Neural networks fail to exploit spatial structure in input data such as images. A $n \times n$ image must be reshaped into a one-dimensional vector of size n^2 in order to train neural networks, which therefore neglects spatial information about the image. This issue required the design of a new class of neural network architectures, namely convolutional neural networks (CNNs), that account for locality in the input data (Lecun and Bengio 1995; LeCun et al. 2015).

CNNs are one of the most powerful techniques at present, yielding breakthrough results in image recognition (He et al. 2015; Krizhevsky et al. 2012; Simonyan and Zisserman 2014; Szegedy et al.

2014), natural language processing (Clark et al. 2018; Devlin et al. 2018; Jozefowicz et al. 2016), object detection (Diba et al. 2017; Ouyang et al. 2016), reinforcement learning, and many other fields. In Chapter 5, we will train a CNN to learn dark matter halo formation from cosmological simulations.

Architecture

Convolutional neural networks are defined as a feed-forward neural network with the addition of at least one convolutional layer prior to the fully-connected layers. In the context of a convolutional neural network, a convolution is a linear operation that involves a dot product between a set of weights and the input summed by a bias term, similar to a traditional neural network. The difference is that the weights are in a two-dimensional matrix, called a *convolutional kernel* (or, *filter*), as CNNs were designed to work on images. CNNs are not just limited to 2-D images but can be generalized to volumetric data, or any n-dimensional data, simply by adopting three-dimensional, or n-dimensional, convolutional kernels. In Chapter 5, we will make use of 3-D CNNs to learn from volumetric N-body simulations. For simplicity, I will describe convolutions assuming the inputs are two-dimensional images but the same arguments apply to higher-dimensional data. The size k of the $k \times k$ filter is usually much smaller than the size of the input data, where typical values of k are 3, 4, and 5. The filter is then shifted systematically to each filter-sized patch of the input data, left to right, top to bottom. Crucially, the weights of the filter remain the same as the filter is applied to different parts of the input image. Therefore, if a given filter is designed to detect a specific type of feature in the input, it will discover that feature anywhere in the image. This capability is commonly referred to as *translation invariance* and is a powerful property when interested in whether a certain feature is present, independent of where it is located in the input. This is one of the main reasons why convolutional neural networks are particularly suited for images.

The output from a single multiplication of the filter with the input is a single value, but as the filter is applied to different patches of the image, the end result is a two-dimensional matrix called a *feature map*. Each pixel in the feature map indicates the strength of the detected feature in different regions of the input image. Typically, a convolutional layer is composed of several feature maps, each constructed using a different convolutional filter, so that multiple features can be extracted from the image in a single convolutional layer. Similar to fully connected layers, each value in the feature maps is passed through a non-linear activation function, as for example a ReLU (Nair and Hinton 2010).

The size of a feature map is controlled by three hyperparameters that have to be set prior to training: the number of convolutional filters, the stride, and amount of zero-padding. The number of convolutional filters determines the number of features to be learnt at any given layer. The stride is the number of pixels by which to slide the filter across the image when performing the convolution. For example, if the stride is one, the filter is moved one pixel at a time across the image, whereas if the stride is two, the filter slides over two pixels at a time. With zero-padding, one pads the image with zeros around the border in order to center the filter on elements at the edge of the image. With appropriate zero padding around the image, and sliding the convolution filter with a stride of 1, the output map will be of the same size as the input. Instead, one can decide to reduce the size of the output map by two by choosing a stride of 2. A detailed review of the arithmetic of convolutional layers can be found in Dumoulin and Visin (2016).

Convolutional layers are often followed by *pooling layers*, which reduce the dimensionality of a feature map by taking the average (*average-pooling*) or the maximum value (*max-pooling*) in small, usually 2×2, regions of the feature maps. They act separately on each feature map, meaning that from a set of N feature maps when obtains the same number of N pooled maps. The effect of the pooling layer is to produce lower-resolution feature maps, which are less sensitive to small changes in the position of the feature in the image compared to the higher-resolution feature maps returned by the convolutional layer. This property is commonly referred to as *local translation invariance*.

Moreover, it is common to add a *batch-normalization layer* (Ioffe and Szegedy 2015), which normalizes the inputs of a batch by first subtracting the batch mean and dividing by the batch standard deviation and then rescaling and shifting the normalized values using two parameters γ and β, which are learnt during backpropagation. This layer is usually placed in between the convolutional layer and the non-linear activation function. Its success is usually attributed to the fact that it reduces the negative impact of changes in the distribution of layer inputs caused by updates to the network parameters in the preceding layers. This effect is known as *internal covariate shift*. However, the exact reasons for its effectiveness are still a matter of debate (Santurkar et al. 2018).

Choosing the exact network architecture is often problem-specific, thus requiring extensive numerical experimentation and intuition. More complicated architectures may be better at capturing complex correlations in the data and learning relevant patterns, but may also lead to overfitting by fitting spurious patterns of the training data. There have also been numerous works that move beyond the simple deep, feed-forward neural network architectures, which for example incorporate "skip connections" that allow information to directly propagate to a hidden or output layer, bypassing

intermediate ones (He et al. 2015). By adding skip connections, the network avoids training for the layers that are not useful and that do not add value in overall accuracy. This idea is particularly helpful in cases where adding more layers to a deep learning model leads to a higher training error.

Training

Similar to regular deep neural networks, convolutional neural networks are trained for a number of epochs, each consisting of a forward pass, where the input passes through the network and reaches the output layer, and a backward pass, where gradients are backpropagated and all the weights in the convolutional layers are updated according to the negative gradient of the loss function. The updates are usually suppressed by multiplying the gradients by a small number α, known as the *learning rate*, which takes values between 0 and 1. The learning rate controls how quickly the model descends towards the minimum of the loss and is one of the most important hyperparameters in the network. If the learning rate is set too low, training will progress very slowly towards the minimum, as only tiny updates are made to the weights each time. If the learning rate is set too high, it can cause drastic changes to the weights, leading to divergent behaviour in the loss function.

The training data can be divided into one or more subsets, called *batches*, which are forward- and backward- propagated through the network independently. In this way, weights are updated after each batch. The size of the batches is a hyperparameter which must be set prior to training. If all training samples are contained in a single batch rather than sub-divided into multiple batches, the learning algorithm is called *batch gradient descent*. If instead each batch consists of a single sample or a subset of the training data, the learning algorithm is called *stochastic gradient descent* or *mini-batch gradient descent*, respectively. One epoch is therefore made of N forward and backward passes for each of the N batches.

Deep convolutional neural networks contain a large number of hyperparameters to be set before training, making their tuning a challenging task. These involve architecture-specific parameters, such as the number of layers (including convolutional, pooling, batch-normalization and fully-connected layers), the number of epochs, the gradient descent optimizer, the learning rate and the choice of loss function, as well as layer-specific parameters. For convolutional layers, these include the size of the kernels, the choice of non-linear activation function, the amount of stride and zero-padding; for pooling layers, the amount of down-sampling and the type of pooling (max or average); for fully-connected layers, the number of neurons and the choice of non-linearity. The large number of hyperparameters to tune in convolutional neural networks makes a fully

grid-based optimization search infeasible. Most of these choices require a large number of numerical trial-and-error stages, which can be done in a systematic way to explore the sensitivity of the learning to the various parameters and the degeneracies between the parameters.

Representation learning

One important and powerful aspect of deep learning algorithms lies in its ability to learn relevant features from the raw data, a task known as *representation learning*. This is in contrast to most of the other machine learning algorithms, such as random forests or gradient boosted trees, which require pre-processing the data into a selected set of features used for training. The hierarchical structure of deep learning models is thought to be crucial to their ability to represent complex features. For example, in convolutional neural networks, the first layers learn local low-level features, as for example edges in images, which are then combined by subsequent layers of the network into more global, higher-level features (Le et al. 2011). This is because each pixel in a convolutional layer is only a function of the $k \times k$ pixels in the previous layer that are contained inside the $k \times k$ kernel of the convolutional filter. Since typically $k \in \{3, 5, 7\}$, the algorithm will only learn local features. As more convolutional layers are stacked on top of each other, the region of the input that any given pixel is a function of increases. The size of this region at any specific layer is called *receptive field* of the layer. The receptive field increases layer-by-layer making each layer sensitive to features at increasingly larger scales. In this way, both local and global information propagate through the network.

Knowledge extraction

Going back to the accuracy vs. interpretability trade-off, deep CNNs provide the most striking example of extremely high accuracy and low interpretability. Model interpretability is currently lacking for CNNs (see e.g. Zhang and Zhu 2018 for a review on the topic of interpretability in deep learning); there is very little insight into the internal operation of these complex models, or how they achieve such good performance. The feature maps generated by each convolutional layer in the network, show patterns in the data which are hard to relate to human-interpretable quantities. In addition to this, the complex network of inter-connected neurons of the CNN also makes it difficult to quantify how individual features then map onto the resulting predictions. The building blocks of interpretability can be summarized in terms of understanding (i) the functionality of the different hidden layers, (ii) the relationship and interconnection between neurons and (iii) how features are

assembled throughout the network up to the final prediction (Olah et al. 2018).

The machine learning community has tried to answer such questions by developing tools based on feature visualization (Olah et al. 2017; Springenberg et al. 2014; Zeiler and Fergus 2014), attribution (Fong and Vedaldi 2017; Selvaraju et al. 2016; Simonyan et al. 2013; Zhou et al. 2015), and dimensionality reduction (van der Maaten and Hinton 2008). Feature visualization tools usually project a model's learnt feature map back to the pixel space using a technique called deconvolution. The aim is to provide insight into what types of features deep neural networks are learning at specific layers in the model. However, due to the large number of feature maps in the network, these methods are usually only applied to few, low-level feature maps in the first layers and so can only provide limited insight. Moreover, qualitative evaluation of such maps can be difficult to turn into quantitative statements about the learnt features. The most common approach for attribution techniques is a *saliency map*, a colormap that highlights pixels of the input image that most caused the output classification (Selvaraju et al. 2016; Zhou et al. 2015). These methods are also limited in that they do not take into account correlations between pixels, or that they only display pixels of the input that are relevant to a single class. Finally, dimensionality reduction methods such as t-SNE have been used (not only for neural networks) to reduce the dimensionality of high-dimensional features into lower dimensions, while trying to preserve the characteristics of the data. Machine learning interpretability is an active area of research, and we expect further improvements in existing and novel interpretation techniques in the future.

A deeper understanding of powerful machine learning algorithms directly ties to the potential for *knowledge extraction*. In physics, this means extracting the underlying physics of a given problem from the machine learning results. We outline future work on knowledge extraction in Chapter 6, where we plan to modify the convolutional neural network architecture used in Chapter 5 to one from which it is possible to extract physical knowledge of cosmological structure formation.

3

Machine learning dark matter halo formation: a binary classification framework

3.1 Abstract

We train a machine learning algorithm to learn cosmological structure formation from N-body simulations. The algorithm infers the relationship between the initial conditions and the final dark matter haloes, without the need to introduce approximate halo collapse models. We gain insights into the physics driving halo formation by evaluating the predictive performance of the algorithm when provided with different types of information about the local environment around dark matter particles. The algorithm learns to predict whether or not dark matter particles will end up in haloes of a given mass range, based on spherical overdensities. We show that the resulting predictions match those of spherical collapse approximations such as extended Press-Schechter theory. Additional information on the shape of the local gravitational potential is not able to improve halo collapse predictions; the linear density field contains sufficient information for the algorithm to also reproduce ellipsoidal collapse predictions based on the Sheth-Tormen model. We investigate the algorithm's performance in terms of halo mass and radial position and perform blind analyses on independent initial conditions realisations to demonstrate the generality of our results.

3.2 Introduction

Dark matter haloes are the fundamental building blocks of cosmic large-scale structure, and galaxies form by condensing in their cores. Understanding the structure, evolution and formation of dark matter haloes is therefore an essential step towards understanding how galaxies form and ultimately, to test cosmological models. However, this is a difficult problem due to the highly non-linear

nature of the haloes' dynamics. Dark matter haloes originate from random perturbations seeded in the early Universe and grow via mass accretion and mergers with smaller structures throughout their assembly history. N-body simulations provide the only practical tool to compute non-linear gravitational effects starting from an initial random field (e.g. Kuhlen et al. 2012; Springel 2005; Springel et al. 2001).

As discussed in Chapter 1, analytic approximations of structure formation yield useful physical interpretations of these detailed numerical studies. Generally, analytic techniques assume dark matter collapse occurs once the smoothed linear density contrast exceeds a threshold value. Combined with excursion set theory, this ansatz provides a tool to analytically predict the final halo mass of an initially overdense region. This can be used to infer useful quantitites such as the abundance of dark matter haloes in the Universe, or the halo mass function, based on properties of a Gaussian random field alone (Bond and Myers 1996; Bond et al. 1991; Press and Schechter 1974). The halo mass function is the quantity most often used to assess the accuracy of different analytic frameworks against numerical simulations. The original form of the halo mass function proposed by Press and Schechter (1974), although qualitatively correct, is known to underestimate the abundance of the most massive haloes, and overestimate the abundance of the less massive ones. The need for precision mass functions led to modifications of the original halo mass function in the form of parametric functions calibrated with cosmological simulations (Jenkins et al. 2001; Reed et al. 2003; Tinker et al. 2008). Pure analytic extensions of the excursion set ansatz have also been constructed which yield better agreement with numerical simulations (Borzyszkowski et al. 2014; Farahi and Benson 2013; Maggiore and Riotto 2010; Paranjape and Sheth 2012; Sheth et al. 2001). In particular, one widely-used generalization of the Press-Schechter formalism is the peak-patch theory (Bond and Myers 1996), which combines the excursion set picture of Bond et al. (1991) and the 'peaks' picture of Bardeen et al. (1986). Given these successful predictions, the excursion set description has become an accepted physical interpretation of the process of structure formation itself.

We present a machine learning approach to learn cosmological structure formation directly from N-body simulations. The machine learning algorithm is trained to learn the relationship between the initial conditions and final halo population that results from non-linear evolution. Using the resulting initial conditions-to-haloes mapping, we aim to provide new physical insights into the process of dark matter halo formation, and compare with existing interpretations gained from widely investigated analytic frameworks. In contrast to existing analytic theories, our approach does not require prior assumptions about the physical process of halo collapse; the haloes' non-linear

dynamics is learnt directly from N-body simulations rather than approximated by an excursion set model in the presence of a collapse threshold.

We provide the machine learning algorithm with a set of informative properties about the dark matter particles extracted from the initial conditions. Machine learning algorithms are sufficiently flexible to include a wide range of initial conditions properties which may contain relevant information about halo formation, without changing the training process of the algorithm. We choose these properties to be aspects of the initial density field in the local surroundings of the dark matter particles' initial position. By quantifying their impact on the learning accuracy of the algorithm, we can investigate which aspects of the early universe density field contain relevant information on the formation of dark matter haloes. The trained initial conditions-to-haloes mapping can then also be used to predict the mapping for new initial conditions, without the need to run a further simulation.

The highly non-linear nature of dark matter evolution makes it a problem well-suited to machine learning. Machine learning is a highly efficient and powerful tool to learn relationships that are difficult to solve analytically or numerically using standard statistical techniques (Witten et al. 2016). In the context of structure formation, machine learning techniques have also been shown to be effective, for example, in learning the relationship between dark and baryonic matter from semi-analytic models (Agarwal et al. 2018; Kamdar et al. 2016; Nadler et al. 2018).

We choose *random forests* (Breiman 2001; Breiman et al. 1984; see Sec. 2.1.2), a popular algorithm which has been shown to outperform other classifiers in many problems (Caruana and Niculescu-Mizil 2006; Douglas et al. 2011; Lochner et al. 2016; Niculescu-Mizil and Caruana 2005). Random forests also lend themselves to physical interpretation, as they provide measures that allows the user to infer which of the inputs are predominantly responsible for the learning outcomes of the algorithm. Random forests are ensembles of decision trees, each following a set of simple decision rules to predict the class of a sample (Ball and Brunner 2010). The prediction of the random forest is given by the average of the probabilistic predictions of the individual trees, where the variance of the forest predictions is greatly reduced compared to that of a single tree.

To apply this approach, we must turn the process of dark matter evolution into a supervised classification problem. We chose to focus on the simplest case of a binary classification task to illustrate the approach and allow for a cleaner understanding of the physics behind the learning process of the algorithm. We distinguish between dark matter particles which end up in haloes of mass above a threshold, and those which belong either to lower mass haloes or to no halo at all. This defines two classes; the former set of particles belongs to the *IN haloes* class while the latter

forms the *OUT haloes* class. The machine learning algorithm is trained to predict whether the dark matter particles in the initial conditions will end up in IN class haloes or in the OUT class at $z = 0$. The training is performed on an existing N-body simulation where we already know the associated halo for each particle (if any).

The predictive accuracy of the algorithm crucially depends on the choice of features extracted from the initial conditions and used as input to the machine learning algorithm. We first train the random forest with the initial linear density field as features and subsequently add information on the tidal shear field. We are able to quantify the physical relevance of such properties in the halo collapse process, based on their respective impact on the classification performance of the random forest. Our results demonstrate the utility of machine learning in gaining insights into the physics of structure formation, as well as providing a fast and efficient classification tool.

This chapter is organized as follows. We present an overview of the classification pipeline and describe how we extract features from the linear density field and train the machine learning algorithm in Sec. 4.3. In Sec. 3.4 we interpret the classification output and present our results in Sec. 3.5. We then extend the feature set to include the tidal shear field in Sec. 3.6 and discuss the resulting implications. We study the algorithm's performance as a function of halo properties in Sec. 3.7. We perform two blind tests of our pipeline on independent simulations in Sec. 3.8, demonstrating the generality of our results, and finally conclude in Sec. 4.8.

3.3 Method

We trained and tested the random forest with an existing dark-matter-only simulation produced with `P-GADGET-3` (Springel 2005; Springel et al. 2001) and a WMAP5 ΛCDM cosmological model (Dunkley et al. 2009); $\Omega_\Lambda = 0.721$, $\Omega_\mathrm{m} = 0.279$, $\Omega_\mathrm{b} = 0.045$, $\sigma_8 = 0.817$, $h = 0.701$, $n_s = 0.96$. The comoving softening length of the simulation is $\epsilon = 25.6\,\mathrm{kpc}$. The simulations evolve 256^3 dark-matter particles, each of mass $M_\mathrm{particle} = 8.24 \times 10^8\,\mathrm{M}_\odot$, in a box of comoving size $L = 50\ h^{-1}\mathrm{Mpc}$ from $z = 99$ to $z = 0$.[1]

The haloes were identified using the `SUBFIND` halo finder (Springel et al. 2001), a friends-of-friends method with a linking length of 0.2, with the additional requirement that particles in a halo be gravitationally bound. While `SUBFIND` also identifies substructure within haloes, we consider the entire set of bound particles to make up a halo and do not subdivide them further. The simulation

[1] We make use of the Python package `pynbody` (Pontzen et al. 2013) to analyse the information contained in the simulation snapshots.

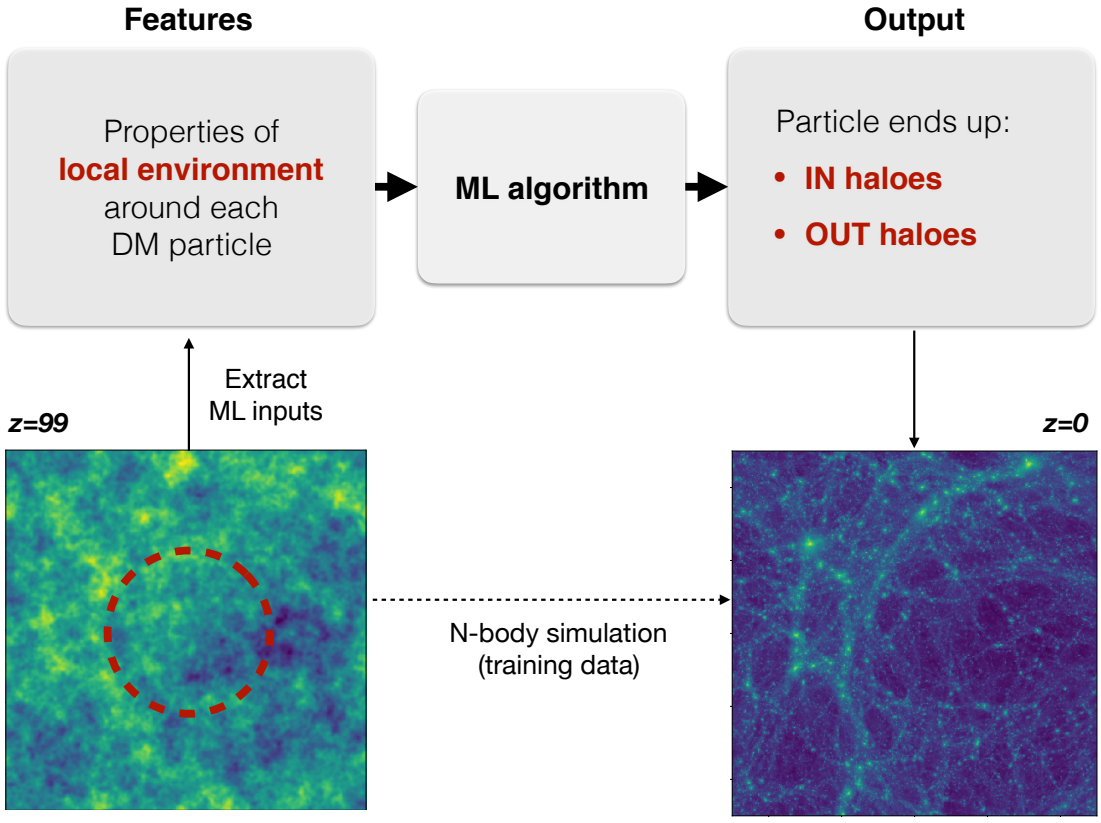

Figure 3.1: An illustration of our binary classification framework. We extract features from the initial conditions of an N-body simulation, describing properties of the local environment around each dark matter particle. Based on these inputs, the machine learning algorithm is trained to predict whether a dark matter particle ends up in the *IN haloes* class or the *OUT haloes* class at $z = 0$, as defined in the text.

contains $18,801$ haloes at $z = 0$, ranging from masses of $\sim 10^9$ $\mathrm{M}_\odot$ to $\sim 10^{14}$ $\mathrm{M}_\odot$.

We used the the final snapshot ($z = 0$) to label each particle with its corresponding class. At $z = 0$, we split the dark matter particles between two classes; *IN haloes* and *OUT haloes*. We chose the IN class to contain all particles in haloes of mass $M \geq 1.8 \times 10^{12}$ $\mathrm{M}_\odot$ at $z = 0$ (401 haloes), and the OUT class to contain all remaining particles, including those in haloes of mass $M < 1.8 \times 10^{12}$ $\mathrm{M}_\odot$ and those that do not belong to any halo.[2] This choice was made in order to split the haloes into the two classes at an intermediate scale within the mass range probed by the simulation. Our pipeline allows the selection of any mass threshold which would ultimately allow us to extend the binary classification to a multi-class one.

[2]The mass scale $M = 1.8 \times 10^{12}$ $\mathrm{M}_\odot$ corresponds to the mass of a particular halo of the simulation and was chosen as the class boundary for convenience.

Each particle, with its associated class label, was traced back to the initial conditions ($z = 99$) where we extracted features to be used as input for the random forest as described below. The random forest was trained based on these input features and the known output class for a training subset of particles. We tested the algorithm using the remaining dark matter particles, where the random forest's class prediction was compared to their respective true class label. The robustness of the algorithm was tested further on independent N-body simulations (Sec. 3.8). An illustration of the binary classification framework is shown in Fig. 3.1.

3.3.1 Density Field Features

Most machine learning algorithms, including random forests, require a *feature extraction* process to extract key properties of the dark matter particles. The classification performance crucially depends on whether or not the chosen features provide meaningful information to allow for a clean separation between the IN and OUT classes.

We extracted machine learning features from the linear density field. This choice was motivated by the work of Press and Schechter (1974) (PS) who developed a model to predict the (comoving) number density of dark-matter haloes as a function of mass based on properties of the linear density field. The ansatz is that a Lagrangian patch will collapse to form a halo of mass M at redshift z if its smoothed linear density contrast exceeds a critical value $\delta_c(z)$. An improved theoretical footing for PS theory was developed by Bond et al. (1991) based on the excursion-set formalism, known as extended Press-Schechter (EPS). The crucial assumption is that the final halo mass corresponds to the matter enclosed in the *largest* possible spherical region with density contrast $\delta_L = \delta_c$. This method yields a halo mass function qualitatively consistent with numerical simulations, suggesting that a useful mapping between Lagrangian regions and final collapsed haloes can be obtained from spherical overdensities. This motivates our choice of machine learning features from the initial linear density field as follows.

We smoothed the density contrast $\delta(\mathbf{x}) = [\rho(\mathbf{x}) - \bar{\rho}]/\bar{\rho}$, where $\bar{\rho}$ is the mean matter density of the universe, on a smoothing scale R,

$$\delta(\mathbf{x}; R) = \int \delta(\mathbf{x}') \, W_{\mathrm{TH}}(\mathbf{x} - \mathbf{x}'; R) \, \mathrm{d}^3 x', \tag{3.1}$$

where $W_{\rm TH}(\mathbf{x}, R)$ is a real space top-hat window function

$$W_{\rm TH}(\mathbf{x}, R) = \begin{cases} \dfrac{3}{4\pi R^3} & \text{for } |\mathbf{x}| \leq R, \\ 0 & \text{for } |\mathbf{x}| > R. \end{cases} \tag{3.2}$$

The convolution (4.3) was carried out in Fourier space, which naturally accounts for the periodicity of simulations. A window function $W(\mathbf{x}, R)$ of characteristic radius R corresponds to a mass scale $M_{\rm smoothing} = \bar{\rho} V(R)$, where in the case of a top-hat window function $V_{\rm TH}(R) = 4/3\pi R^3$. The feature for machine learning then consists of the density contrast smoothed with a top-hat window function of mass scale $M_{\rm smoothing}$ (or, smoothing scale R) centred on the particle's position in the initial conditions.

We repeated the smoothing for 50 mass scales evenly spaced in $\log M$ within the range allowed by the volume and resolution of the simulation box i.e., $3 \times 10^{10} \leq M_{\rm smoothing}/{\rm M}_{\odot} \leq 1 \times 10^{15}$, yielding a set of 50 features per particle. When adopting a smaller number of smoothing mass scales, e.g. 20 or 30 bins, we found a decrease in the performance of the algorithm. On the other hand, we found that using a larger number of smoothing scales did not yield improvement in the classification performance, meaning that 50 smoothing scales were sufficient to capture the relevant information carried by the density field.

In the context of excursion set theory, the density contrast of a particle as a function of smoothing scale is known as a *density trajectory*. Fig. 3.2 shows examples of density trajectories of particles belonging to the true IN and OUT classes. The trajectories describe whether particles are found in overdense or underdense regions as a function of increasing mass scale. As one approaches the largest mass scales probed by the simulation box, the trajectories start to converge to $\delta(x, \infty) = 0$, where the density coincides with the mean density of the Universe. The ensemble of trajectories constitutes the full feature set we used to first train then test the random forest.

3.3.2 Training the random forest

We make use of the random forest implementation in the SCIKIT-LEARN (Pedregosa et al. 2011) Python package. The random forest was trained using a set of $50{,}000$ randomly selected particles from the simulation, each carrying its own set of density features and corresponding IN or OUT class label. The size of the training set was chosen to form a subset of particles representative of the full simulation box. To test for representativeness, we checked the performance of the algorithm for training sets of different sizes and found no improvement for training sets larger than $50{,}000$

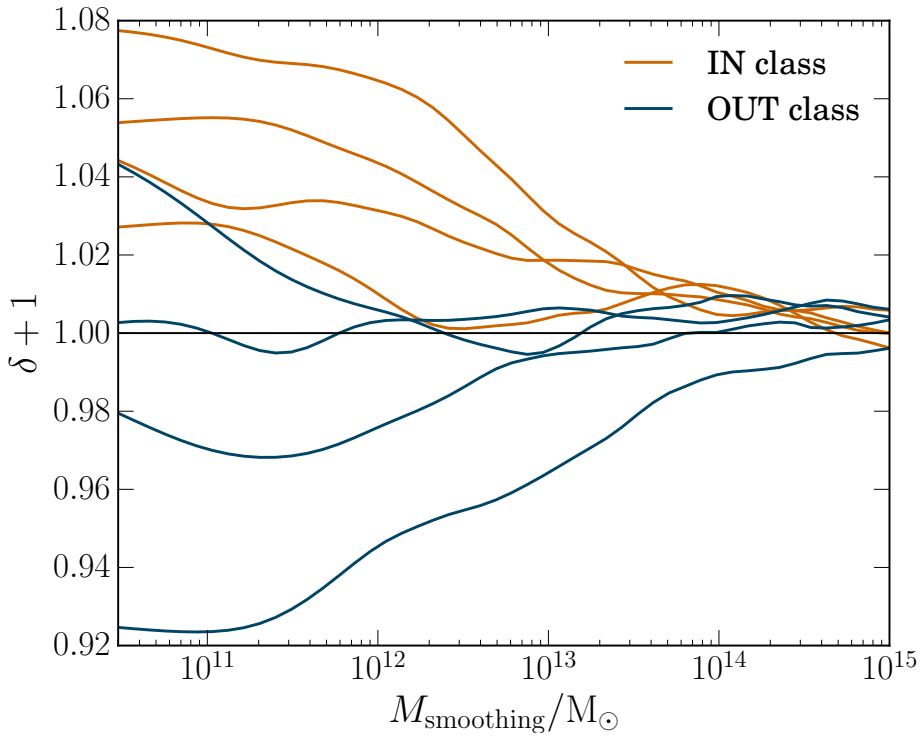

Figure 3.2: Examples of density trajectories corrresponding to particles belonging to the IN and OUT classes. The linear density field is smoothed with a real space top-hat filter centred on each particle's initial position. We calculate the smoothed overdensity δ as the smoothing mass scale M is increased.

particles. Therefore, we concluded that $50{,}000$ randomly selected particles are sufficient to form a training set representative of the full simulation box. The remaining particles in the simulation were used as a test set; the trained random forest predicts the class label of the particles in the test set, which is then compared to the particles' true labels to assess the algorithm's performance. Note also that random forests are robust to correlated features (Breiman 2001), meaning that the high correlation present in our density features does not affect the predictive performance of the algorithm.

Like most machine learning algorithms, random forests have hyperparameters which need to be optimized for a given training set. These include the number of trees and the maximum depth of the forest, the maximum number of particles at the end node of a tree and the size of the subset of features to select at a node split. We used a grid search algorithm combined with k-fold cross validation (Kohavi 1995) to optimize the random forest's hyperparameters, as described in more details in Sec. 2.1.1. In k-fold cross validation, the training set is divided into k equally sized sets where $k-1$ sets are used for training and one is used as a validation set, on which the algorithm is

Table 3.1: Confusion matrix for two classes: Positives and Negatives. We use this to quantify the performance of the machine learning algorithm, where the positives are particles of the IN class and the negatives are particles of the OUT class.

		True Class	
		P	**N**
Predicted Class	**P**	True Positive (TP)	False Positive (FP)
	N	False Negative (FN)	True Negative (TN)

tested. This procedure is repeated k times so that each set is used as a validation set once. For each validation set we evaluate a score based on a chosen scoring metric (here we use the area under the Receiver Operating Characteristic curve, see Sec. 3.4) and average scores over all k validation sets to obtain the final score of a training set. Here, we performed a five-fold cross validation for all combinations of hyperparameters and retained the combination which achieved the best score.

3.4 Interpreting the classification output

A random forest (like most machine learning algorithms) outputs a probabilistic measure of belonging to a class for every particle. For practical use this must be mapped onto a concrete class for each particle. Many approaches exist for such a mapping. For example, Leclercq et al. (2015) proposed a Bayesian decision theory approach, motivated by game theory, to classify the cosmic web into different structure types. We choose to consider different probability thresholds at which a particle is considered to belong to a class. A high probability threshold will contain a very pure sample of particles but also will be incomplete. As the probability threshold decreases, one allows for a more complete set of particles at the expense of including misclassified ones.

Once the probability-to-class mapping is established, we quantify the performance of the algorithm making use of a confusion matrix for binary classification problems as shown in Table 3.1. Throughout this analysis we always take the positives to be particles of the IN class and negatives to be particles of the OUT class. The perfect classifier consists of true positives and true negatives only. A more realistic classifier will include a number of incorrectly classified particles: misclassified positives fall in the false negative category, yielding a loss of *completeness*, and misclassified negatives fall in the false positive category, yielding an increase in *contamination*. We measure the true positive rate (TPR), the ratio between the number of particles correctly classified as positives and the total

number of positives in the data set,

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{3.3}$$

and the false positive rate (FPR), the ratio between the number of particles incorrectly classified as positives and the total number of negatives in the data set,

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}. \tag{3.4}$$

Receiver Operating Characteristic (ROC) curves (Fawcett 2006; Green and Swets 1966; Hilden 1991) are a tool to graphically represent the balance between completeness and contamination at various probability thresholds. A ROC curve compares the true positive rate to the false positive rate as a function of decreasing probability threshold. As one lowers the probability threshold, one allows for a more complete set of IN particles (increase in true positive rate) at the expense of a larger contamination of misclassified particles (increase in false positive rate). The area under the curve (AUC) of a ROC curve is a useful quantity to compare classifiers. The perfect classifier would have an AUC of 1, whereas a random assignment of classes would obtain an AUC of 0.5. Typically, algorithms are considered to be performing well if AUC ≥ 0.8.

We use ROC curves and AUCs to evaluate and compare the performance of the random forest for different feature sets (Sec. 3.5 & 3.6), different halo mass and radial position ranges (Sec. 3.7) and different simulations (Sec. 3.8).

3.5 Density field Classification

Figure 3.3 shows the ROC curve for the density feature set resulting from classifying all particles in the simulation that were not used for training the random forest. The random forest achieves an AUC score of 0.876.

In order to assess whether machine learning can learn as much as human-constructed models, we wish to compare its performance to existing theories. In particular, the EPS formalism motivated our choice of density features and has been demonstrated to infer approximately correct number densities of collapsed haloes from a Gaussian random field (Bond et al. 1991). Although EPS is commonly used to predict the dark matter halo mass function, we make use of it to predict an independent set of class labels for the test set particles and compare their accuracy to that of the

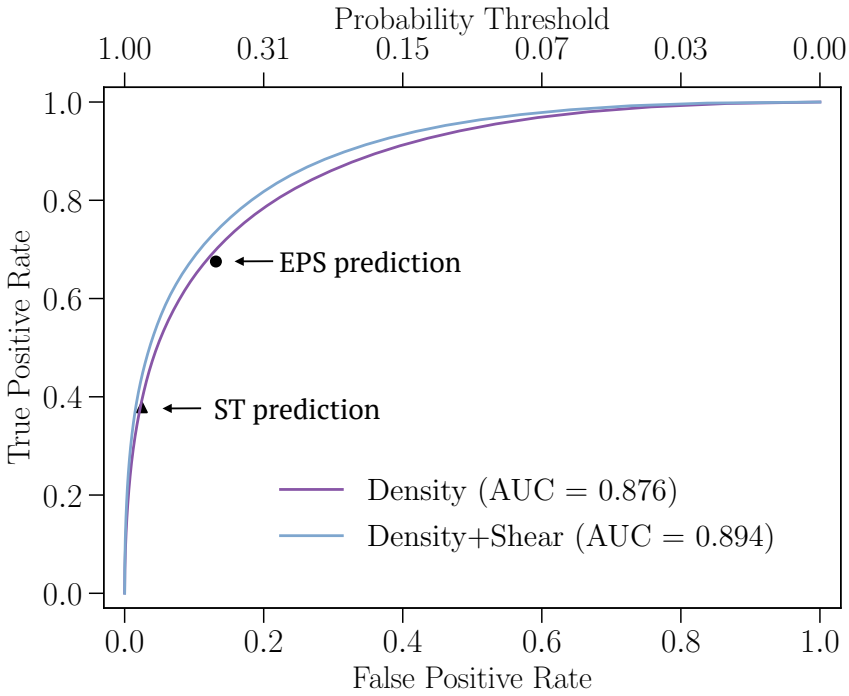

Figure 3.3: ROC curves for the density feature set and the combined shear and density feature set. The machine learning algorithm is able to learn the information contained in the density trajectories to match the EPS prediction. The ST prediction represents an extension of standard excursion set developed by Sheth and Tormen (1999), which adopts a moving collapse barrier motivated by tidal shear effects. The comparison between the two ROC curves shows little improvement in the test set classification once information on the shear field is added. The ST analytic prediction also does not provide an overall improvement compared to the EPS prediction; the false positive rate (or, contamination) decreases at the expense of decreasing the true positive rate (or, completeness). The machine learning algorithm is able to recover the ST analytic prediction when presented with information on the density field alone by altering the probability threshold.

machine learning predictions.

Following EPS, the fraction of haloes of mass M is equivalent to the fraction of density trajectories with a first upcrossing of the density threshold barrier $\delta_{\rm th}$ at mass scale M. We take the density threshold to be the spherical collapse threshold adopted by Bond et al. (1991): $\delta_{\rm th}(z) = (D(z)/D(0))\,\delta_{\rm sc}$, where $\delta_{\rm sc} \approx 1.686$. The predicted halo mass of each particle is given by the smoothing mass scale of the particle's first upcrossing. We then assign to each particle an IN or OUT label depending on whether its predicted halo mass falls in the mass range of the IN or OUT class. We emphasize that the labels inferred from the EPS framework are independent from the predictions of the random forest.

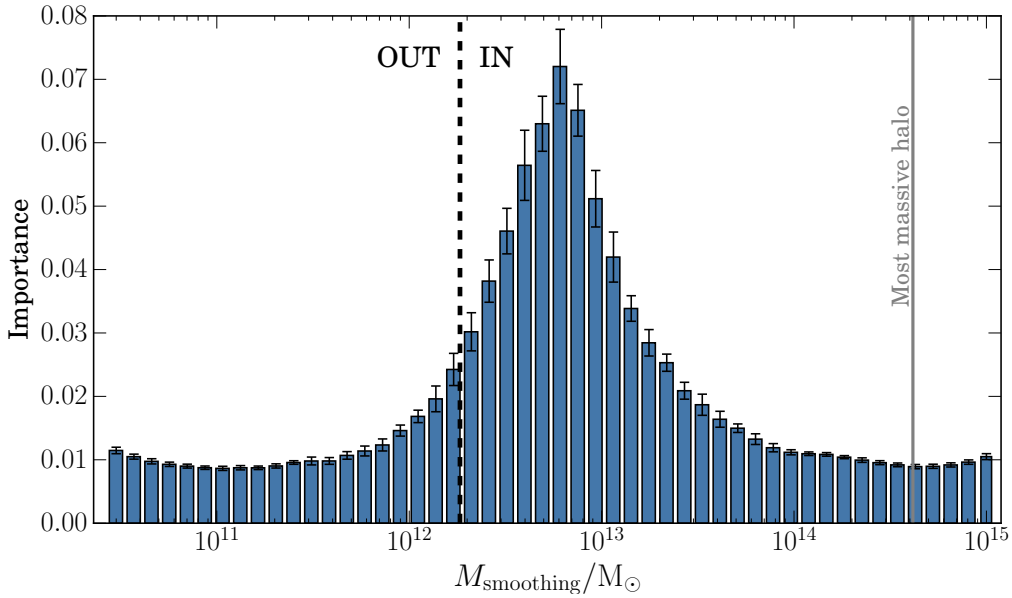

Figure 3.4: The importance ranking of the density features, shown as a function of their smoothing mass scales. The most relevant information in the training of the random forest comes from the density contrast smoothed at mass scales 10^{12} – 10^{13} $\mathrm{M}_\odot$ scales, within the mass range of the IN class haloes. The largest halo mass in the simulation is marked by a grey line.

We plot in Fig. 3.3 the resulting true positive rate and false positive rate inferred from the EPS predicted labels and find that the EPS prediction lies on the ROC curve of the random forest. In other words, the random forest is able to 'learn' EPS and the EPS results correspond to a $\sim 42\%$ probability threshold on the ROC curve. Machine learning adds the flexibility to trade contamination for completeness along the ROC curve as we vary the probability threshold. Instead, EPS results in a single point in true positive rate-false positive rate space since it gives a single prediction for each particle rather than a probability associated with a class.

3.5.1 Physical Interpretation

The algorithm's performance depends on whether or not the input features contain relevant information to separate particles between classes. For example, the ideal feature would split a set of particles into two pure sets, each containing only particles of one class. By contrast, irrelevant features are not able to distinguish between classes, yielding a poor class separation in the two resulting sets. Therefore, we can determine which features contain the most information in mapping particles into the correct halo mass range, based on their ability to separate classes when training the random forest.

There are many metrics designed to measure the relevance of the inputs to a machine learning algorithm; here we use *feature importances* (Louppe et al. 2013). The importance of a feature X is a weighted sum of the impurity decrease[3] at all nodes t where the feature is used, averaged over all trees T in the forest:

$$\mathrm{Imp(X)} = \frac{1}{N_T} \sum_T \sum_{t \in T} p(t) \Delta i(t), \tag{3.5}$$

where N_T is the number of trees, $p(t)$ is the fraction of particles reaching node t and $\Delta i(t)$ is the impurity decrease, i.e. the difference in entropy between the parent node and the child nodes. See Sec. 2.1.3 for further details, where feature importances were introduced.

We calculate the relative importances in the density feature set to find the most relevant features in distinguishing between the IN and OUT classes. Fig. 3.4 shows the relative importance of each density feature as a function of its smoothing mass scale. The importances are normalized such that the sum of all importances is 1 and the errors are computed by training the random forest multiple times, each with a randomly drawn set of training particles. The largest halo mass in the simulation is marked by a grey line. We find that most of the information lies in mass ranges of 10^{12} – 10^{13} $\mathrm{M}_\odot$, just above the boundary between the IN and OUT classes.

3.6 Adding the tidal shear tensor

Peaks in Gaussian random fields are inherently triaxial (Bardeen et al. 1986; Doroshkevich 1970). Therefore, extensions of the standard spherical model were made in order to incorporate the dynamics of ellipsoidal collapse. The impact of the tidal shear on properties of collapsed regions has been extensively studied (Bond and Myers 1996; Lin et al. 1965; Sheth and Tormen 1999; Sheth et al. 2001). Sheth and Tormen (1999) (ST) have studied how ellipsoidal collapse modifies the mass function of dark matter haloes in the excursion set formalism. Spheres are distorted into an ellipsoid due to tidal shear effects and the collapse time of a halo therefore depends explicitly on the ellipticity and prolateness of the tidal shear field.

We extended the original density feature set to incorporate additional information on the local tidal shear field around particles. We studied the impact on the halo classification performance and quantified the shear's relevance in the training process via the feature importances. The advantage

[3]We use Shannon entropy to measure the impurity at a node $i_E(t) = -\sum_{i=1}^{c} p(j,t) \log_2 p(j,t)$, where $p(j,t)$ is the proportion of particles that belong to class j at node t and c is the total number of classes.

of studying tidal shear effects with machine learning is that these can be straightforwardly translated into features and used as input to the same machine learning algorithm. On the other hand, analytic models usually require incorporating approximations to the tidal shear within the excursion set formalism. In general, any potentially relevant physical property can be added in the form of a feature without adding complexity to the algorithm.

We will first describe how we constructed features from the tidal shear field, then present the classification results of the full density and shear feature sets.

3.6.1 Tidal shear features

The deformation tensor is given by the Hessian of the gravitational potential

$$D_{ij} = \frac{\partial^2 \Phi}{\partial x_i \partial x_j}, \tag{3.6}$$

where $\Phi(\mathbf{x})$ is the peculiar gravitational potential at position $\mathbf{x}$ and is related to the density contrast via Poisson's equation $\nabla^2 \Phi = \delta$.

The ordered eigenvalues of D_{ij}, $\lambda_1 \geq \lambda_2 \geq \lambda_3$, can be re-parametrized in terms of the ellipticity, e, and prolateness, p (Bond and Myers 1996):

$$e = \frac{\lambda_1 - \lambda_3}{2\delta}, \tag{3.7}$$

$$p = \frac{\lambda_1 - 2\lambda_2 + \lambda_3}{2\delta}, \tag{3.8}$$

where $\lambda_1 + \lambda_2 + \lambda_3 = \delta$ and δ is the smoothed overdensity used as a density feature. In order to minimize redundancy between the features, we removed the density dependence from the ellipticity and prolateness. We computed the eigenvalues of the traceless deformation tensor, known as the tidal shear tensor, $t_i = \lambda_i - \delta/3$, now satisfying $t_1 + t_2 + t_3 = 0$. The ellipticity and prolateness in terms of the traceless eigenvalues t_i take the form

$$e_t = t_1 - t_3, \tag{3.9}$$

$$p_t = 3\,(t_1 + t_3)\,. \tag{3.10}$$

For each particle we assigned two new features e_t and p_t evaluated at each smoothing mass scale. Therefore, the original 50–dimensional feature set of density contrasts was augmented to a 150–dimensional feature set given by the density contrast, ellipticity and prolateness. To test

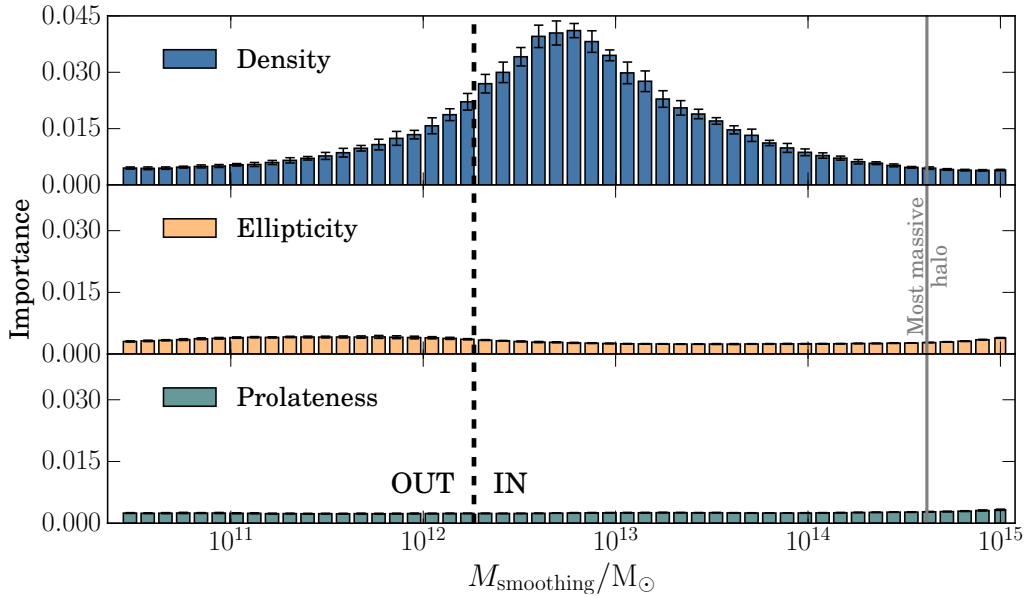

Figure 3.5: Relative importance of the density features (*upper panel*), ellipticity features (*middle panel*) and prolateness features (*lower panel*) in the full shear and density feature set. The density features are more relevant than the ellipticity and prolateness features. This confirms that the shear field adds little information in distinguishing whether particles will collapse in haloes of mass above the class boundary mass scale or not, compared with the density field.

the robustness of random forests to a high-dimensional feature space, we used PCA to reduce the 150–dimensional feature set to a 10–dimensional space retaining 98% of the information contained in the original feature set. We found identical predictive performance, meaning that random forests are robust to a 150–dimensional feature set.

3.6.2 Results

The ROC curve of the density and shear feature set is overplotted in Fig. 3.3. We find that adding information on the tidal shear tensor shows little improvement compared to the case of the density-only feature set. We find an improvement of only 2% in the AUC of the ROC curve. Fig. 3.5 demonstrates the low impact of the shear features in the classification process. The three panels show the relative importance in the training process of the random forest of the density, ellipticity and prolateness features as a function of smoothing mass scales. The most relevant features are the density contrasts smoothed on mass scales in the range 10^{12} – 10^{13} $\mathrm{M}_{\odot}$, similar to what was found in the case of the density-only feature set (Fig. 3.4). The distributions of the density importances in the two feature sets are consistent despite minor variations in the peak and variance of the

distributions. The changes are due to the change in the range of hyperparameters when increasing the dimensionality of the feature set from 50 to 150 features. The ellipticity and prolateness have low feature importance scores confirming that the information they contain is irrelevant to the training process of the machine learning algorithm compared with that of the density field.

As with the density feature set, we can compare the machine learning predictions to existing analytic predictions based on the same set of properties of the initial conditions. The ST formalism provides a prescription to predict the final halo mass of a particle based on the density field and the shear field, which we can use to compare to the machine learning output.

ST accounts for the effect of the shear field in the context of the excursion set formalism by adopting a moving collapse barrier rather than the spherical collapse barrier adopted by Bond et al. (1991). The ST collapse barrier $b(z)$ varies as a function of the mass variance $\sigma^2(M)$ and is given by

$$b(z) = \sqrt{a}\delta_{\rm sc}(z)\left[1 + \left(\beta\frac{\sigma^2(M)}{a\delta_{\rm sc}^2(z)}\right)^{\gamma}\right], \tag{3.11}$$

where $\delta_{\rm sc}(0) \approx 1.686$, the parameters $\beta = 0.485$ and $\gamma = 0.615$ incorporate an approximation to ellipsoidal dynamics, and $a = 0.707$ is a normalisation constant. These values are the best-fit parameters found in Sheth and Tormen (1999). The predicted halo mass of each particle follows the excursion-set framework as for the EPS case; the largest mass scale at which the particle's trajectory up-crosses the collapse barrier in Eq. (3.11) gives the predicted halo mass.

The triangle labelled "ST prediction" in Fig. 3.3 shows the true and false positive rates predicted by ST. In our study, the ST formalism does not yield an absolute improvement to EPS theory; the false positive rate decreases at the expense of a decrease in the true positive rate. Therefore ST predicts a less contaminated but more incomplete set of IN class particles compared to EPS, corresponding to a probability threshold of 73% on the ROC curve. We find that the random forest is able to reproduce the ST result with both the density-only feature set and the shear and density feature set. This shows that there is sufficient information in the density field for the random forest to match the analytic ST prediction.

Overall, we find that shear effects do not contain additional physical information to improve the classification output of the random forest. The learning process of the algorithm is predominantly driven by the local overdensity around dark matter particles and unaffected by the surrounding tidal shear. The analytic ST prediction, interpreted as an improvement to standard EPS due to the inclusion of tidal shear effects, can be reproduced by the random forest when trained on the density field only. In conclusion, these results show that the physical processes leading to dark matter halo

formation for our choice of mass scale splitting the two classes are insensitive to tidal shear effects in the initial conditions.

3.7 Classification dependence on halo mass and radial position

We now investigate how properties of particles such as the position within a halo and the halo mass affect the accuracy of classification when the algorithm is trained on density features only. To do this we split the test particles into categories based on their radial and halo mass properties to study their respective classification performance.

First, we subdivided particles of the IN class into three mass ranges: particles in *cluster*-sized haloes ($1 \times 10^{14} \leq M_{\rm halo}/{\rm M}_{\odot} \leq 4 \times 10^{14}$), particles in *group*-sized haloes ($1 \times 10^{13} \leq M_{\rm halo}/{\rm M}_{\odot} < 1 \times 10^{14}$) and particles in *galaxy*-sized haloes ($1.2 \times 10^{12} \leq M_{\rm halo}/{\rm M}_{\odot} < 1 \times 10^{13}$). We combined each of these subsets in turn with all the OUT particles to form three distinct test sets.

The ROC curves for the three mass range categories of haloes are shown in the right panel of Fig. 3.6, where the ROC curve of the full original test set is shown for comparison (dashed line). We find that particles in cluster-sized haloes reach an AUC of 0.913, whilst particles in group-sized haloes and galaxy-sized haloes are increasingly more difficult to classify. We overplotted the ST (triangles) and EPS (dots) predictions for each halo mass category of particles, again showing results consistent with those of the machine learning algorithm.

It is likely that the decrease in performance as a function of halo mass is a result of the choice of mass scale used to split haloes into classes, $M = 1.8 \times 10^{12}\ {\rm M}_{\odot}$. This was a necessary step in order to define the two classes of the binary classification problem. Haloes of mass just above and below the IN/OUT mass boundary belong to different classes although they originate from Lagrangian regions with similar properties reflecting their similarity in mass. Therefore, the closer haloes of different classes are in mass, the harder it is for the random forest to distinguish whether their particles belong to one class or the other. Fig. 3.7 further demonstrates that haloes of mass approaching the IN/OUT mass boundary from above and below contain a larger fraction of misclassified particles. In the upper (lower) panel, we show the false positive (negative) rate i.e., the ratio of misclassified OUT (IN) particles over all particles contained in each halo mass bin, for 4 different probability thresholds. The true halo mass of each particle is shown on the horizontal axis in terms of its distance from the IN/OUT mass boundary. We find that the false positive and negative rates increase for particles in haloes of mass approaching the IN/OUT mass boundary.

We next investigated possible correlations between the particles' position within the haloes and

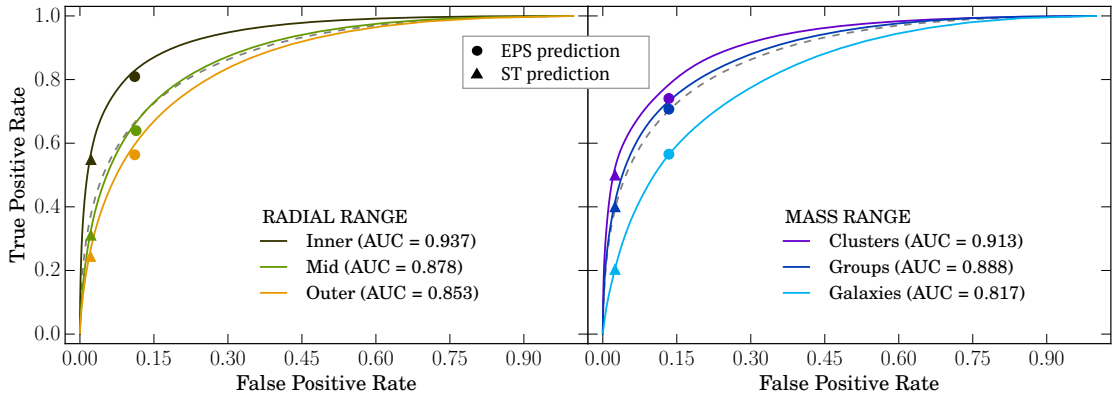

Figure 3.6: *Left panel*: The IN class particles are split into inner ($r/r_{\rm vir} \leq 0.3$), mid ($0.3 < r/r_{\rm vir} \leq 0.6$) and outer ($0.6 < r/r_{\rm vir} \leq 1$) radial ranges according to their distance from the centre of the halo. The ROC curves for each category show that the classification performance improves for particles closer to the halo's centre of mass. *Right panel*: The IN class particles are split into cluster-sized ($1 \times 10^{14} \leq M_{\rm halo}/{\rm M}_{\odot} \leq 4 \times 10^{14}$), group-sized ($1 \times 10^{13} \leq M_{\rm halo}/{\rm M}_{\odot} < 1 \times 10^{14}$) and galaxy-sized ($1.2 \times 10^{12} \leq M_{\rm halo}/{\rm M}_{\odot} < 1 \times 10^{13}$) haloes and the ROC curves show the random forest's performance in classifying each category. Particles in higher mass haloes are increasingly better classified by the random forest. The ROC curve of the full test set of particles is shown as a dashed line in both panels for comparison. The EPS and ST predictions, labelled by dots and triangles respectively, are also overplotted for each halo mass and radial position category.

the random forest's classification performance. Here, we subdivided particles of the true IN class into three radial ranges, subject to their radial position in the halo with respect to the halo's virial radius $r_{\rm vir}$. We defined particles in the *inner radial* range ($r/r_{\rm vir} \leq 0.3$), particles in the *mid radial* range ($0.3 < r/r_{\rm vir} \leq 0.6$) and particles in the *outer radial* range ($0.6 < r/r_{\rm vir} \leq 1$). Similar to the mass range study, each subset of haloes was combined with all the OUT class particles from the original set to form three distinct sets.

The left panel of Fig. 3.6 shows the ROC curves for the three radial categories, together with that of the original test set again shown for comparison (dashed line). Particles in the innermost regions of haloes are the best classified by the random forest, achieving an AUC of 0.937 which is greater than that obtained when classifying *all* particles in the simulation. The classification performance of the random forest decreases as we move from the halo's centre-of-mass towards the virial radius.

We first tested whether the decrease in performance when classifying particles of the outer radial range was due to under-representativeness in the training set. Indeed, if the training particles of the outer radial range are not representative of the entire simulation, the classifier's performance

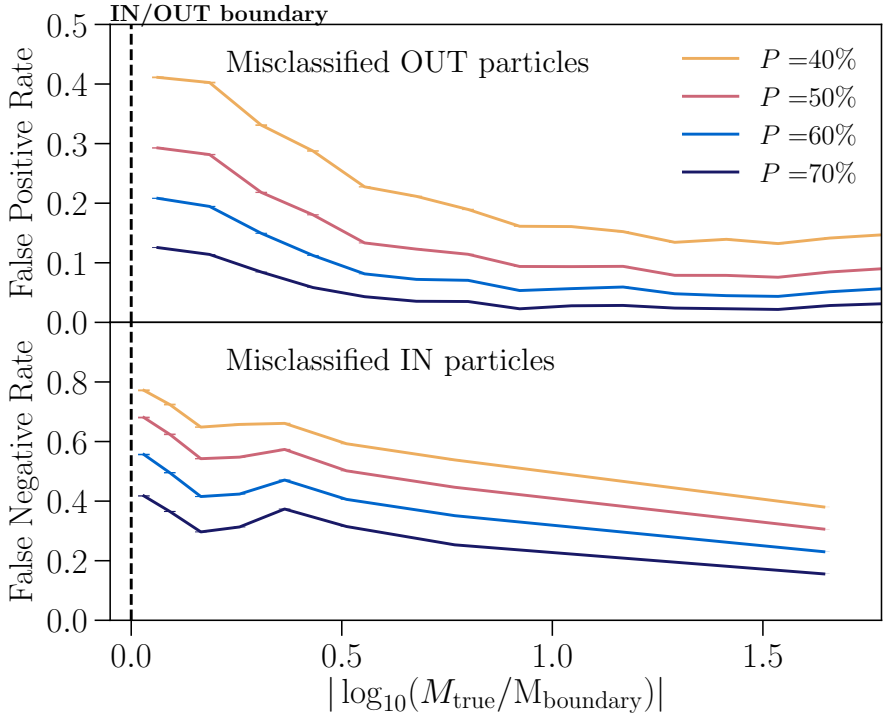

Figure 3.7: Fraction of misclassified particles in haloes of each mass bin range, where the halo mass bins are labelled as a function of their distance from the IN/OUT boundary mass scale. The upper (lower) panel shows the fraction of misclassified OUT (IN) particles i.e., the false positive (negative) rate in each mass bin. We consider four distinct probability thresholds for assigning a particle's (IN or OUT) class, where higher thresholds imply lower contamination. The misclassification rate increases as the true mass approaches the classification boundary for all choices of the completeness-to-contamination trade-off.

on the outer radial range test set would be strongly affected. To test this, we re-trained the machine learning algorithm with a training set containing equal number of particles for each radial range category. We found identical ROC curves and AUCs as in the left panel of Fig. 3.6, therefore excluding the possibility that the higher misclassification rate of outer radial range particles is due to non-representativeness in the training set.

One other possible reason may be that particles living in outer regions of haloes are more likely to have been affected by late-time halo mergers, tidal stripping or accretion events. Therefore, the final halo mass prediction for such particles is the result of a more complicated dynamical history involving these late-time effects. Conversely, particles near the halo's centre-of-mass are less sensitive to the halo's assembly history and their final halo mass prediction correlates more strongly with the local overdensity in the initial conditions. This hypothesis could be verified by

adding features sensitive to the particles' dynamical history (for instance a particle's initial distance to the nearest density peak) and testing whether this information improves the classification of particles located at the boundary of the halo's virial region. In addition to this, the further particles are from the centre of haloes, the closer they are to the boundary between the IN and OUT classes, where particles are harder to classify for the machine learning algorithm. This also translates into a larger uncertainty in the halo mass prediction for particles at the edge of haloes compared to those in the innermost regions of haloes. As a result, the overall uncertainty in the halo mass predictions of centre-of-mass particles is smaller than for particles in the outskirts of haloes. This result is also consistent with excursion set predictions, where ST demonstrated that centre-of-mass particles provide a better estimate of the final halo mass compared to inferences made from the full ensemble of particles in the simulation. To confirm this, we overplotted the EPS (dots) and ST (triangles) predictions for the three radial test sets in the left panel of Fig. 3.6, demonstrating that analytic formalisms also perform increasingly well for particles that are close to the halo's centre-of-mass. The machine learning algorithm again shows its ability to match the excursion set predictions at fixed probability thresholds for each radial range category.

For completeness, we also explored the misclassification rate of OUT particles that do not belong to any halo. We find that overall these particles have very low misclassification rates compared to particles in haloes. For example, if we consider probability thresholds of 70%, 60%, 50% and 40% to assign particles to the IN class (as in the upper panel of Fig. 3.7), the fraction of misclassified over all particles that don't belong to haloes is 2.45%, 4.3%, 6.58% and 10.11%, respectively. Therefore, the OUT particles predicted by the random forest form a highly pure and complete set.

In conclusion, we find that the best classified categories of particles are those which are further away from the classification boundary, both in terms of mass and radius: particles in the most massive and least massive haloes in the simulation; particles in the innermost regions of haloes; and those furthest away in voids. We further tested whether the addition of the tidal shear information could improve the classification performance of poorly classified particles, such as those in the outskirts of haloes and in galaxy-sized haloes. We find no significant improvement in the classification performance of such particles, other than the 2% improvement found for the whole ensemble and reflected in each mass and radial category.

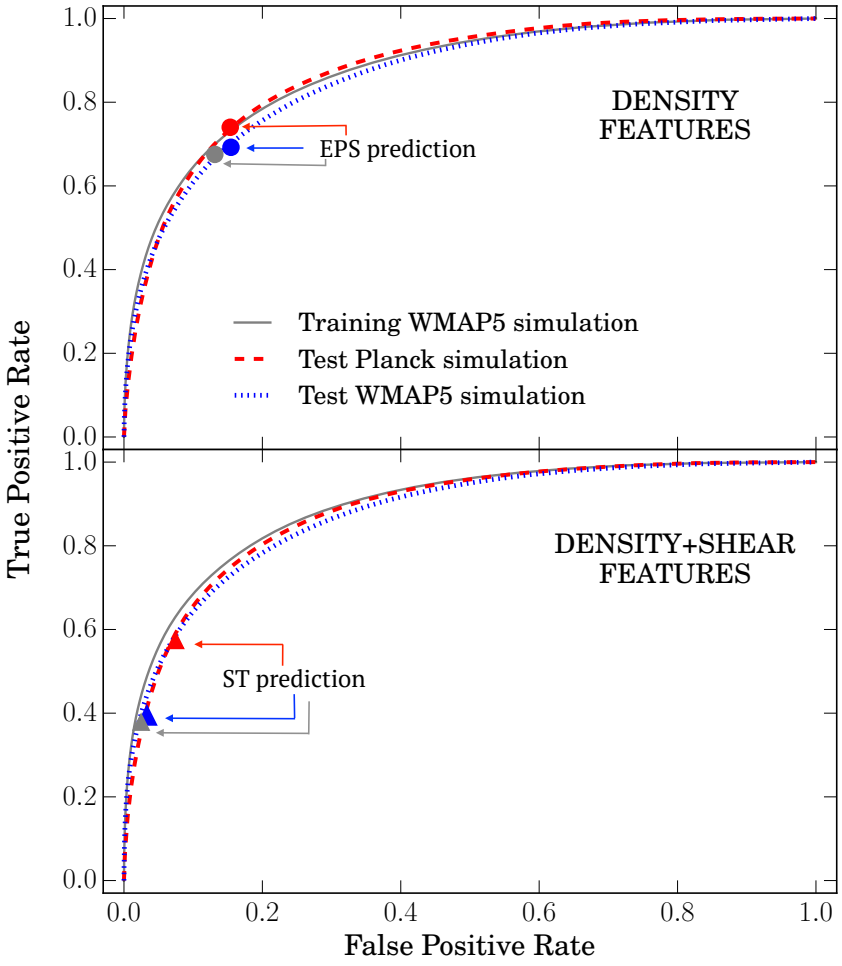

Figure 3.8: We perform a blind test of the trained machine learning algorithm on two independent N-body simulations; a different realisation of the WMAP5 cosmology used in the training simulation, and a realisation of a *Planck* cosmological model. The ROC curves are consistent in all three simulations for both the density feature set and the density and shear feature set, with differences in the AUCs of order $\sim 1\%$. The EPS and ST predictions in each simulation match the machine learning performance at different probability thresholds, such that the ST formalism always predicts a less contaminated but more incomplete set of IN particles. These blind tests demonstrate the robustness of the results from a machine learning algorithm trained on one simulation, and applied to different realisations of the same cosmology or realisations of different cosmologies.

3.8 Blind tests on independent simulations

Up to this point we have trained and tested the machine learning algorithm on a single dark-matter-only simulation. To test whether the machine learning algorithm trained on one simulation also gives robust results for different N-body simulations without re-training, we performed blind tests

of our pipeline on two independent simulations from the one used for training.

The first independent test simulation (W-Test) is a different realisation of the same WMAP5 ΛCDM cosmology adopted in the training simulation, for a box of also same size and resolution (see Sec. 4.3). The second independent test simulation (P-Test) is a realisation of a different cosmological model, a *Planck* ΛCDM cosmology[4](Planck Collaboration et al. 2016) in a box of comoving size $L = 50\,\mathrm{Mpc}$ containing $N = 512^3$ particles. Moreover, in the P-Test simulation we identify haloes at $z = 0$ using the Amiga Halo Finder (AHF) (Gill et al. 2004; Knollmann and Knebe 2009), instead of the SUBFIND halo finder used in both the training simulation and the W-Test simulation. This allows us to simultaneously test the sensitivity of the machine learning algorithm to the choice of halo finder. For each test simulation, we extracted the input features from the initial conditions and used the pre-trained machine learning algorithm to predict the class labels of the simulations' dark matter particles.

In Fig. 3.8 we compare the performance of the machine learning algorithm for the independent W-Test and P-Test simulations with that of the test set of particles in the training simulation. The upper panel shows the ROC curves obtained from predictions based on the density features only, whilst the lower panel shows the case of density and shear features. The machine learning algorithm produces consistent ROC curves in all three simulations for both feature sets. The P-Test simulation yields a difference in AUC with the training simulation of 0.2% for the density-only feature set and 1.1% for the density and shear feature set. For the W-Test simulation, the AUC difference with the training simulation is of 1.3% for the density-only feature set and 1.6% for the density and shear feature set. Such differences between the test and training simulations are consistent with uncertainties in the AUC due to statistical noise.

The EPS and ST predicted labels are calculated from the first upcrossings of each simulation's respective particles' trajectories. In all three simulations, the machine learning algorithm is able to match the analytic predictions at different probability thresholds, such that the ST formalism consistently predicts a less contaminated but more incomplete set of IN class particles. For the W-Test simulation, the EPS and ST predictions match the machine learning predictions at probability thresholds of 41.5% and 74.5% respectively, differing only slightly to the 42.8% and 74.7% probability thresholds of the training simulation. For the P-Test simulation, the match to the EPS and ST predictions is found at the lower probability thresholds of 40% and 56%, respectively. This is because the change in cosmological parameters in the *Planck* simulation results in a slightly lower EPS collapse barrier and a significantly lower ST collapse barrier compared to those in a WMAP5

[4]The cosmological parameters are $\Omega_\Lambda = 0.6914$, $\Omega_\mathrm{m} = 0.3086$, $\Omega_\mathrm{b} = 0.045$, $\sigma_8 = 0.831$, $h = 0.6727$, $n_s = 0.96$.

cosmological setting. Therefore, trajectories in the P-Test simulation upcross the collapse barriers at larger smoothing mass scales, resulting in more complete but also less pure sets of predicted IN particles. The change in completeness and contamination is such that both the ST and EPS predictions still match the machine learning ROC curves of the P-Test simulation, but for lower probability thresholds than the WMAP5 simulations.

We conclude that the mapping learnt by the algorithm on one simulation can be generalized to different simulations based on the same or different cosmological parameters, without the need for re-training, and that the results are insensitive to simulation settings.

3.9 Conclusions

We have presented a machine learning approach to investigate the physics of dark matter halo formation. We trained the algorithm on N-body simulations, from which it learns to predict whether regions of an initial density field later collapse into haloes of a given mass range. This generated a mapping between the initial conditions and final haloes that would result from non-linear evolution, without the need to adopt halo collapse approximations. Our approach provided new physical insight into halo collapse, in particular in understanding which aspects of the initial linear density field contain relevant information on the formation of dark matter haloes.

We provided the algorithm with a set of properties describing the local environment around dark matter particles. By studying the performance of the algorithm in response to different inputs, insights can be gained into the physics relevant to dark matter halo formation. When the algorithm was trained on spherical overdensities from the linear density field, we found that it matched predictions based on EPS theory. When providing the algorithm with additional information on the tidal shear field (motivated by ellipsoidal collapse approximations), the classification performance of the machine learning was not enhanced. We showed that, for the mass threshold considered in our classification problem, the Sheth-Tormen ellipsoidal collapse model can be recovered from spherical overdensities alone, with predictions that differ from those of EPS theory only in the completeness-to-contamination trade-off. By performing blind analyses of our pipeline, we confirmed the generality of our results for independent initial conditions realisations and variations in cosmological parameters. We conclude that the linear density field contains sufficient information to predict the formation of dark matter haloes at the accuracy of existing spherical and ellipsoidal collapse analytic frameworks. Therefore, the fact that the ST halo mass function improves the EPS halo mass function may not be due to the addition of tidal shear information, but rather some other physical effects captured by

calibrating the free parameters in the halo mass function to the simulations.

While the focus of this paper has been on the density field and tidal shear field, any additional property of interest can be extracted from the initial conditions and used as input to the same machine learning algorithm. This allows for straightforward extensions of this work to investigate which additional physics in the early universe contributes to the formation of dark matter haloes. One other natural extension is to turn the binary classification problem presented in this chapter into multi-class classification or regression problems. This will be the focus of Chapter 4.

Acknowledgements

LLS thanks Nina Roth for providing one of the simulations used in this work and for useful discussions. LLS was supported by the Science and Technology Facilities Council. HVP was partially supported by the European Research Council (ERC) under the European Community's Seventh Framework Programme (FP7/2007-2013)/ERC grant agreement number 306478- CosmicDawn. AP was supported by the Royal Society. ML acknowledges support from the SKA, NRF and AIMS. This work was partially enabled by funding from the UCL Cosmoparticle Initiative.

4 Machine learning dark matter halo formation: a regression framework

4.1 Abstract

We present a generalization of our recently proposed machine learning framework, aiming to provide new physical insights into dark matter halo formation. We investigate the impact of the initial density and tidal shear fields on the formation of haloes over the mass range $11.4 \leq \log(M/M_{\odot}) \leq 13.4$. The algorithm is trained on an N-body simulation to infer the final mass of the halo to which each dark matter particle will later belong. We then quantify the difference in the predictive accuracy between machine learning models using a metric based on the Kullback-Leibler divergence. We first train the algorithm with information about the density contrast in the particles' local environment. The addition of tidal shear information does not yield an improved halo collapse model over one based on density information alone; the difference in their predictive performance is consistent with the statistical uncertainty of the density-only based model. This result is confirmed as we verify the ability of the initial conditions-to-halo mass mapping learnt from one simulation to generalize to independent simulations. Our work illustrates the broader potential of developing interpretable machine learning frameworks to gain physical understanding of non-linear large-scale structure formation.

4.2 Introduction

The evolution of dark matter haloes is determined by a series of complex, non-linear physical processes involving smooth mass accretion and violent mergers with smaller structures. For decades, N-body simulations have been used to model the non-linear evolution of haloes (e.g. Springel et al.

2005). Alongside these, simpler approximate analytic models of halo collapse can provide qualitative understanding of the results of numerical simulations. For example, extended Press-Schechter (EPS) theory and Sheth-Tormen (ST) theory are two widely accepted analytic frameworks used to infer statistical properties of dark matter haloes starting from an initial Gaussian random field. EPS theory is based on the assumption that halo collapse occurs spherically, once the smoothed linear density contrast exceeds a certain threshold (Bond et al. 1991; Press and Schechter 1974). The ST formalism is an extension of EPS theory to an ellipsoidal collapse model which accounts for the effect of tidal shear forces around initial peaks (Bond and Myers 1996; Doroshkevich 1970). These models require restrictive assumptions about the physical process of halo collapse; the haloes' non-linear evolution is approximated as spherical and ellipsoidal respectively, and formulated using excursion set theory.

Machine learning provides a tool that is well suited to modelling cosmological structure formation, given its ability to learn non-linear relationships. In fact, machine learning tools have already proved useful in the context of structure formation in, for example, distinguishing between cosmological models (Merten et al. 2019) or constructing mock dark matter halo catalogues (Berger and Stein 2019). However, understanding the inner workings of machine learning models remains a challenge. Developing tools to turn "black-box" algorithms into interpretable ones is essential for machine learning applications to physics problems; it will allow us to interpret results in terms of the underlying physics.

In Chapter 3, we proposed a machine learning approach which aims to provide new physical insights into the physics of the early universe responsible for halo collapse. A machine learning algorithm is trained to learn the relationship between the early universe and late-time haloes from N-body simulations. Unlike existing analytic theories, our machine learning approach does not require modelling halo collapse with an excursion set theory; the haloes' non-linear dynamics is learnt directly from N-body simulations. The algorithm's learning is based on properties of the linear initial condition fields surrounding each dark matter particle. Machine learning algorithms are sufficiently flexible to include a wide range of properties of the initial conditions which may contain relevant information about halo formation, without changing the training process of the algorithm. By comparing the predictive performance of the algorithm when provided with different types of inputs, one can gain insights into which aspects of the early universe impact the later formation of dark matter haloes.

In Chapter 3, we focused on the simplest case of a binary classification problem; the algorithm classified dark matter particles into two classes, depending on whether or not they will form

haloes above a specified mass threshold at $z = 0$. Contrary to existing interpretations of the Sheth-Tormen ellipsoidal collapse model, we found that the tidal shear field does not contain additional information over that contained in the density field about whether haloes will form above and below a mass threshold $M_{\rm th} = 1.8 \times 10^{12}$ $\rm M_{\odot}$. However, these conclusions were limited to this single mass threshold.

The aim of this work is to extend our machine learning framework to investigate haloes across a wider range of final mass. In practice, we train a machine learning algorithm to predict the value of the final mass of the halo to which each particle will belong. This is now a regression problem since the algorithm's prediction consists of a continuous variable, rather than a class label. We compare the halo mass predictions resulting from two machine learning models, trained on different sets of inputs: one on information about the initial linear density field only, and the other on both density and tidal shear information. The inputs to the algorithm, known as *features* in machine learning terms, are the same as those adopted in Chapter 3. We are able to quantify the relevance of the information contained in the tidal shear relative to that in the density field by comparing the predictions resulting from one model with the other. In this work, we mainly focus on the formation of haloes at $z = 0$, but also verify that our conclusions hold for haloes at higher redshifts.

The chapter is organized as follows. We describe the method in Sec. 4.3, starting with an overview of the pipeline. We then introduce the machine learning algorithm adopted in this work and describe its training and testing procedure. We present the halo mass predictions in Sec. 4.4, including a study of the algorithm's performance as a function of halo properties. We introduce a metric to make a quantitative comparison of machine learning models in Sec. 4.5. We further test the generality of our results on independent simulations in Sec. 4.7, and finally conclude in Sec. 4.8.

4.3 Method

In this paper we made use of six dark-matter-only simulations produced with `P-GADGET-3` (Springel 2005; Springel et al. 2001) and a WMAP5 ΛCDM cosmological model[1] (Dunkley et al. 2009). Adopting an updated set of cosmological parameters (e.g. from Planck Collaboration et al. 2018a) is not necessary for the purpose of this work. We call the simulations *sim-#*, where $\# \in [1, 6]$. Each simulation is based on a different realization of a Gaussian random field drawn from the initial power spectrum of density fluctuations. All simulations consist of a box of comoving size

[1]The cosmological parameters are $\Omega_{\Lambda} = 0.721$, $\Omega_{\rm m} = 0.279$, $\Omega_{\rm b} = 0.045$, $\sigma_8 = 0.817$, $h = 0.701$ and $n_s = 0.96$.

$L = 50\ h^{-1}\mathrm{Mpc}$ and $N = 256^3$ dark-matter particles evolving from $z = 99$ to $z = 0$.[2]

Dark matter haloes were identified at $z = 0$ using the SUBFIND halo finder (Springel et al. 2001), a friends-of-friends method with a linking length of 0.2, with the additional requirement that particles in a halo be gravitationally bound. We took the entire set of bound particles that make up a halo and did not consider substructure within haloes. The resolution and volume of the simulation limit the resulting range of halo masses; the lowest mass halo has $M = 2.6 \times 10^{10}\ \mathrm{M}_\odot$ and the highest mass one $M = 4.1 \times 10^{14}\ \mathrm{M}_\odot$.

To train and test the machine learning algorithm, we first established the link between the initial and final state of each dark matter particle in the simulations. We used the final snapshots ($z = 0$) to label each particle with the logarithmic mass of the halo to which that particle belongs. Particles that do not collapse into haloes make up $\sim 50\%$ of all particles in the simulations, implying a strong class imbalance between particles not in resolved haloes and those spread across haloes of different mass scales. Training the algorithm to learn such an imbalanced mapping strongly degraded the accuracy of the predictions for particles within resolved haloes. Since our goal is to derive insight into resolved physics, we chose to restrict our analysis to the subset of particles that collapse into resolved haloes at $z = 0$. Out of these, each particle, with its logarithmic halo mass label, was then traced back to the initial conditions where we extracted features to be used as input to the machine learning algorithm.

The algorithm was trained and tested independently on the six different simulations. This yielded six different machine learning models of the same underlying mapping, allowing us to estimate the statistical significance of our results. For each simulation, the algorithm was trained based on the input features to logarithmic halo mass mapping of a training subset of particles. The remaining dark matter particles in the simulation were then used to test the algorithm's predictions against their respective true logarithmic halo mass. We will initially present the results from *sim*-1, but we will draw the final conclusions based on the results from all six simulations.

4.3.1 Gradient Boosted Trees

We used *gradient boosted trees* (Freund and Schapire 1997; Friedman 2001, 2002), a machine learning algorithm combining multiple regression decision trees into a single estimator. A regression decision tree is a model for predicting the value of a continuous target variable by following a simple set of decision rules inferred from the input features. Since individual trees generally over-fit

[2] We made use of the Python package `pynbody` (Pontzen et al. 2013) to analyse the information contained in the simulation snapshots.

the training data, they are often combined together to form a more robust ensemble estimator. There are two main approaches to combine decision trees; *bagging* and *boosting*. Bagging estimators are effective at decreasing variance, but have no effect on the bias. Instead, boosting can reduce both the bias and the variance contributions to the error in the predictions (Schapire et al. 1998) by means of its iterative aggregation of trees. We chose to use boosting estimators, as the bias and variance of the predictions in our dataset both contribute to the predictive error.

The principles of gradient boosted trees were discussed in detail in Sec. 2.1.2. At its core, gradient boosted trees work by combining boosting with gradient-descent optimization. Trees are added iteratively to the ensemble according to the negative gradient of the loss function with respect to the ensemble's predictions. Effectively, subsequent trees correct for the mistakes made by the previous trees in the ensemble.

In addition to the predictive power of this algorithm, gradient boosted trees also allow for very high interpretability of their learning procedure. This is a common feature amongst ensembles of decision trees. We made use of the *feature importances* metric (Louppe et al. 2013; see Sec. 2.1.2) to measure the relevance of each input feature in training the algorithm to predict the correct target variable. This is a crucial aspect of our framework; it allows us to determine which features are most informative in mapping particles to the correct final halo masses. The importance of the j-th feature X_j from a single tree t of the ensemble is given by

$$\mathrm{Imp}_t\left(X_j\right) = \sum_{n \in \{n \text{ is split on feature } X_j\}} \frac{N_n}{N_t}\left[p - \frac{N_{n_R}}{N_n} p_R - \frac{N_{n_L}}{N_n} p_L\right] \tag{4.1}$$

where N_t, N_n, N_{n_R}, N_{n_L} are the total number of samples in the tree t, at the node n, at the right-child node n_R and at the left-child node n_L, respectively. The sum in the equation is over all n nodes where the feature X_j makes the split. The impurity p is given by the choice of splitting criterion, which in our case is the mean squared error. The final importance of feature X_j given by the ensemble of T trees is the normalized sum over the importances from all trees,

$$\mathrm{Imp}(X_j) = \frac{\sum_{t=1}^{T} \mathrm{Imp}_t\left(X_j\right)}{\sum_{j=1}^{J} \mathrm{Imp}\left(X_j\right)}. \tag{4.2}$$

We used the `LightGBM` (Ke et al. 2017) implementation of gradient boosted trees released by Microsoft.

4.3.2 Machine learning Features

A *feature extraction* step is required amongst most machine learning algorithms, including gradient boosted trees, to extract key properties of the dark matter particles and use them as input to the algorithm. Following Chapter 3, we used two properties of the linear density field in the local environment around dark matter particles: the overdensity and the tidal shear computed within spheres of different mass scales centred at each dark matter particle's initial position. These choices were motivated by existing analytic frameworks which provide models to predict the final mass of a halo based on similar properties of the linear density field. EPS theory argues that a spherical patch will collapse to form a halo at redshift z if its average linear density contrast $\delta_L(z)$ exceeds a critical value $\delta_c(z)$, hence motivating our choice of spherical overdensities. The final mass of the halo corresponds to the matter enclosed in the *largest* possible spherical region with density contrast $\delta_L = \delta_c$. The ST framework motivated our choice of tidal shear information. In their approach, the collapse time of a halo depends explicitly on the ellipticity and prolateness of the tidal shear field, as well as on spherical overdensities. Using these properties as machine learning features will allow us to compare the predictions to those from analytic theories based on the same input properties and test the interpretation of these models.

We now briefly discuss how the machine learning features were constructed from the density and tidal shear fields, referring the reader to Chapter 3 for further details. We smoothed the density contrast $\delta(\mathbf{x}) = [\rho(\mathbf{x}) - \overline{\rho}_m] / \overline{\rho}_m$, where $\overline{\rho}_m$ is the mean matter density of the universe, on a smoothing scale R,

$$\delta(\mathbf{x}; R) = \int \delta(\mathbf{x}') \, W_{\rm TH}(\mathbf{x} - \mathbf{x}'; R) \, \mathrm{d}^3 x', \tag{4.3}$$

where $W_{\rm TH}(\mathbf{x} - \mathbf{x}', R)$ is a real space top-hat window function which takes the form

$$W_{\rm TH}(\mathbf{x} - \mathbf{x}', R) = \begin{cases} \dfrac{3}{4\pi R^3} & \text{for } |\mathbf{x} - \mathbf{x}'| \leq R, \\ 0 & \text{for } |\mathbf{x} - \mathbf{x}'| > R. \end{cases} \tag{4.4}$$

We repeated the smoothing for 50 smoothing mass scales (which are related to the smoothing scales R via $M_{\rm smoothing} = 4/3\pi\overline{\rho}_m R^3$), evenly spaced in $\log M$ within the range $3 \times 10^{10} \leq M_{\rm smoothing}/\mathrm{M}_\odot \leq 1 \times 10^{15}$.

From each smoothed density contrast field $\delta(\mathbf{x}, R)$, we computed the peculiar gravitational

potential $\Phi(\mathbf{x})$ via Poisson's equation $\nabla^2\Phi = \delta$ and the tidal shear tensor,

$$T^{\alpha\beta} = \left[\frac{\partial^2}{\partial x^\alpha \partial x^\beta} - \frac{1}{3}\delta^{\alpha\beta}\nabla^2\right]\Phi. \tag{4.5}$$

We assigned two shear features to each dark matter particle, the ellipticity e_t and prolateness p_t, following the definition of Bond and Myers (1996)[3],

$$e_t = t_1 - t_3, \tag{4.6}$$

$$p_t = 3\,(t_1 + t_3)\,. \tag{4.7}$$

where t_1 and t_3 are two of the ordered eigenvalues of the tidal shear tensor (the third is not independent since $t_1 + t_2 + t_3 = 0$). The second term on the right hand side of Eq. 4.5 removes the density field from the tidal shear tensor since $\nabla^2\Phi = \delta$, implying minimal redundancy between the information contained in the density features and that of the shear features.

In summary, we constructed two feature sets; the 50-dimensional *density* feature set made of spherical overdensities, and the 150-dimensional *density and shear* feature set made of spherical overdensities, ellipticity and prolateness features. By comparing the predictive performance of the algorithm when trained on the two feature sets, we were able to test whether the addition of tidal shear information yields an improvement in predicting the formation of the final haloes.

4.3.3 Training a gradient boosted tree

For training the gradient boosted trees, we randomly selected $500{,}000$ particles from those that collapse into haloes at $z = 0$, each carrying its own set of features and final halo mass label. No improvement in the machine learning predictions was found as we increased the size of the training set to more than $500{,}000$ particles, implying that this was sufficient to yield a training set representative of the whole simulation. Note that the training set is ten times larger than that used in Chapter 3; the need for a larger training set is expected due to the higher number of degrees of freedom in a regression setup compared to a binary classification one. The remaining particles in the simulation were used as a test set; the gradient boosted trees were trained to predict the final mass of the halo in which each test set particle will end up. The predictions were then compared to the particles' true halo masses to assess the algorithm's performance.

Gradient boosted trees have hyperparameters which must be set prior to training, and which

[3]We use the eigenvalues of the tidal shear tensor to define the ellipticity and prolateness, rather than those of the deformation tensor like in Bond and Myers (1996).

need to be optimized for any given machine learning problem. The main hyperparameters to optimize are the number of trees in the ensemble, a gradient regularization parameter and the maximum depth and number of leaf nodes in a tree. A popular approach for hyperparameter optimization is to grid-search over a specified subset of hyperparameters and select the optimal ones using k-fold cross validation (Kohavi 1995). A disadvantage of this implementation of the method is that training and validation sets are randomly selected subsets of the same training data. Therefore, this procedure is insensitive to noise present in the training data, as this will be shared amongst both training and validation sets. In our problem, constructing validation and training sets from the same simulation may lead to overfitting the training simulation and as a result, the learned map would fail to generalize to different simulations.

To prevent this, we constructed validation sets from the dark matter particles of a different simulation to the one used for training. All simulations were trained using 5 validation sets from *sim*-2, except for *sim*-2 which used the same number of validation sets from *sim*-1. Each set consists of $50,000$ randomly chosen particles. The hyperparameter optimization procedure then followed the standard 5-fold cross validation approach of choosing the set of hyperparameters best performing on the validation data.

4.3.4 The test set particles

In each simulation, the trained gradient boosted trees can be used to predict the final halo mass of all particles in the simulation in the test set. However, we restricted our analysis to a subset of test set particles satisfying two criteria.

First, we found that gradient boosted trees make biased predictions when the true halo mass is near the limits of the mass range probed by the simulation. The predicted masses of particles in the lowest mass haloes are overestimated and those of particles in the highest mass haloes are underestimated. The closer the true halo mass to the hard cut-offs in mass, the larger the bias in the predicted masses. Since we did not want to base our analysis on predictions affected by algorithm-specific biases, we imposed a criterion to exclude dark matter particles whose predictions are dominated by this bias.

The second criterion excludes all particles that belong to the few haloes found in the simulation at the high mass end. The reason for this will become apparent in Sec. 4.5, when we compare the predicted and true number of particles within bins of halo mass. At the high mass end, there are only a few haloes and therefore a few discrete masses in the training set. Therefore, we adopted

a second criterion that excludes particles with an associated mass label in the range where the shot noise in the expected number of haloes within bins of logarithmic mass is higher than a given threshold.

In practice, these two criteria were implemented as follows. Let us denote $M^i_{\rm predicted}$ and $M^i_{\rm true}$ as the predicted and true halo mass of the i-th particle, respectively[4]. We split the true halo masses of all test particles in k evenly-spaced intervals of logarithmic mass. In each bin, we computed the bias b_k and variance σ^2_k defined as

$$b_k = \frac{1}{J_k}\sum_{j=1}^{J_k}\left[M^j_{k,\rm predicted} - M^j_{k,\rm true}\right], \tag{4.8}$$

$$\sigma^2_k = \frac{1}{J_k}\sum_{j=1}^{J_k}\left|M^j_{k,\rm predicted} - \overline{M}^j_{k,\rm predicted}\right|^2. \tag{4.9}$$

where $M^j_{k,\rm predicted}$ and $M^j_{k,\rm true}$ are the predicted and true halo mass of particle j, $\overline{M}^j_{k,\rm predicted}$ is the mean of the predicted halo masses and J_k is the total number of particles in the k-th bin. This yielded our first criterion; we excluded from the analysis all particles in bins where $b^2_k \geq \sigma^2_k$.

For the second criterion, we first computed the expected number of haloes in each mass bin k, N_k,

$$N_k = V\int_{M_k}^{M_{k+1}} \frac{dn}{dM'}\,dM' \tag{4.10}$$

where V is the volume of the box and $\frac{dn}{dM}$ is the number of haloes of mass M per unit volume per unit interval in M. The latter can be parametrized by the universal functional form

$$\frac{dn}{dM} = f(\sigma)\frac{\overline{\rho}_m}{M}\frac{d\ln\sigma^{-1}}{dM}, \tag{4.11}$$

where $\overline{\rho}_m$ is the cosmic mean matter density and $\sigma^2(M)$ is the mass variance of the linear density field smoothed with a top-hat window function on scale $R(M)$. We adopted the function $f(\sigma)$ predicted by Sheth and Tormen (1999) as it provides a good enough approximation of our simulation's mass function at $z = 0$ and is given by

$$f(\sigma) = A\sqrt{\frac{2a}{\pi}}\left[1 + \left(\frac{\sigma^2}{a\delta_c^2}\right)^p\right]\frac{\delta_c}{\sigma}\exp\left[-\frac{a\delta_c^2}{2\sigma^2}\right], \tag{4.12}$$

where $A = 0.3222$, $a = 0.707$, $p = 0.3$ and $\delta_c = 1.686$. Finally, our second criterion imposed that all

[4]Note that the predicted halo masses were computed without excluding any particles from the training set.

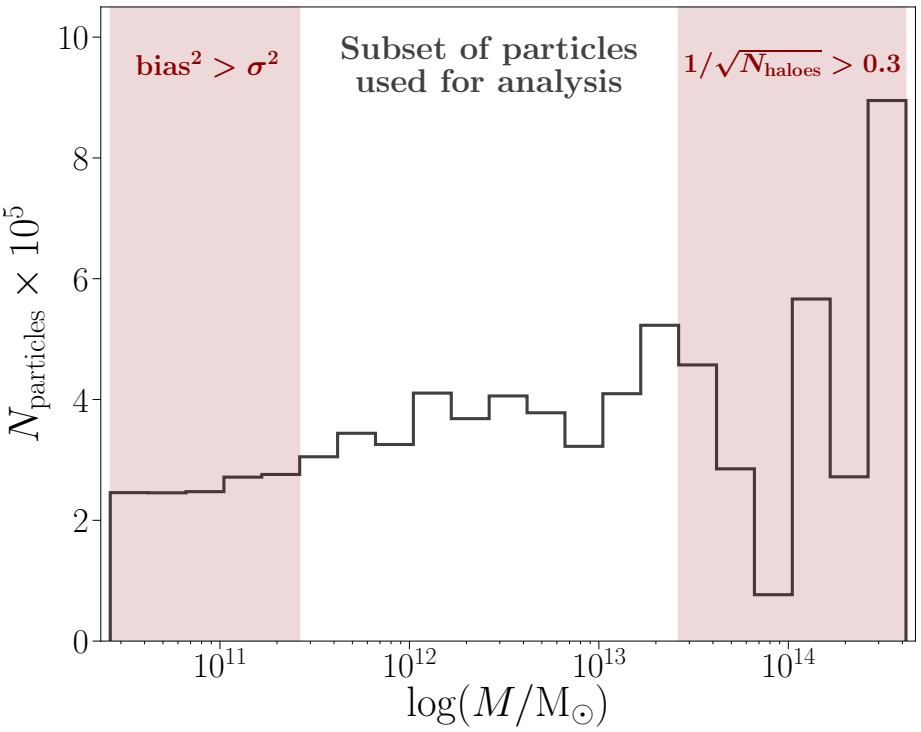

Figure 4.1: All particles in haloes, which were not used for training, were split into k halo mass intervals of width $\Delta \log(M/{\rm M}_\odot) = 0.2$. We excluded from the analysis particles within the k-th mass bins where either of the following criteria are satisfied: (1) the bias in the predictions exceeds the variance i.e., $b_k^2 > \sigma_k^2$, (2) the theoretical number of haloes is smaller than a given threshold i.e., $1/\sqrt{N_{k,\rm haloes}} > 0.3$. Criterion (1) is set to exclude particles in mass bins near the mass limits imposed by the simulation, where the gradient boosted tree makes biased predictions. Criterion (2) is set to exclude mass ranges with small number of haloes. As a result, the particles used for the analysis in all simulations are those in haloes in range $11.4 \leq \log(M/M_\odot) \leq 13.4$.

particles with halo mass label within mass bins where the expected Poisson noise in N_k exceeds 30% i.e., $1/\sqrt{N_k} > 0.3$, were excluded from the analysis.

In summary, the subset of particles from the test set which we retained for our analysis is given by those particles belonging to haloes in k mass bins where the conditions $1/\sqrt{N_k} \leq 0.3$ and $b_k^2 < \sigma_k^2$ are simultaneously satisfied. Both criteria are subject to the choice of bin width defining the k bins; we chose $\Delta \log(M/{\rm M}_\odot) = 0.2$.[5] The criteria were applied to all simulations, for the same choice of bin width. In all simulations, this implied that we retained particles in haloes of mass within the range $11.4 \leq \log(M/M_\odot) \leq 13.4$ for our analysis. Fig. 4.1 shows the *sim*-1 distribution of test set particles in haloes per logarithmic mass intervals, where the shaded regions indicate the mass

[5]The width is chosen in order to be left with at least ten logarithmic mass bins, after applying the criteria.

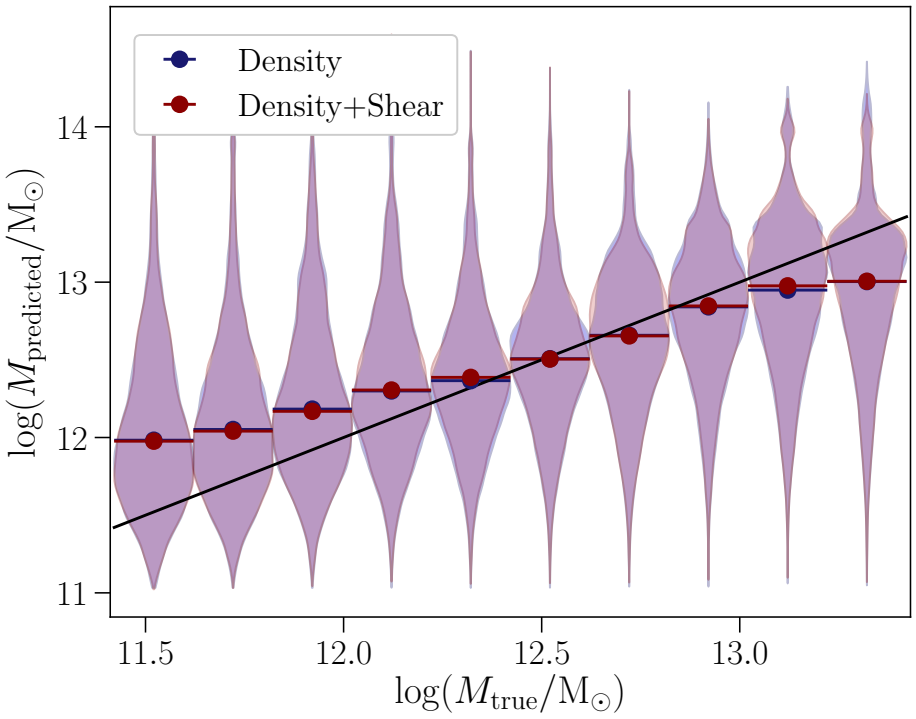

Figure 4.2: Distributions (and their medians) obtained with the predicted halo masses of particles within bins of width $\Delta \log(M/{\rm M}_\odot) = 0.2$, defined by their true logarithmic halo mass. The distributions are in the form of violin plots i.e., box plots whose shapes indicate the distribution of mass values. Within each bin, we compare the distributions predicted by the two machine learning models; one based on density features alone and the other based on both density and shear features. These are near-identical, meaning that there is no qualitative improvement resulting from providing the algorithm with additional information about the tidal shear field.

ranges excluded from the analysis.

4.4 Halo mass predictions

Figure 4.2 compares the machine learning predictions with the true halo masses of the test set particles in *sim*-1. We show the distributions obtained with the predicted halo masses of particles within bins defined by their true logarithmic halo mass. These are shown as violin plots i.e., box plots whose shapes indicate the distribution of mass values. The dots represent the medians of the predicted distributions as a function of the medians within each true mass interval. We compare the distributions resulting from two distinct machine learning models; one trained on the density feature set and the other on the density and shear feature set. We find near-to-identical

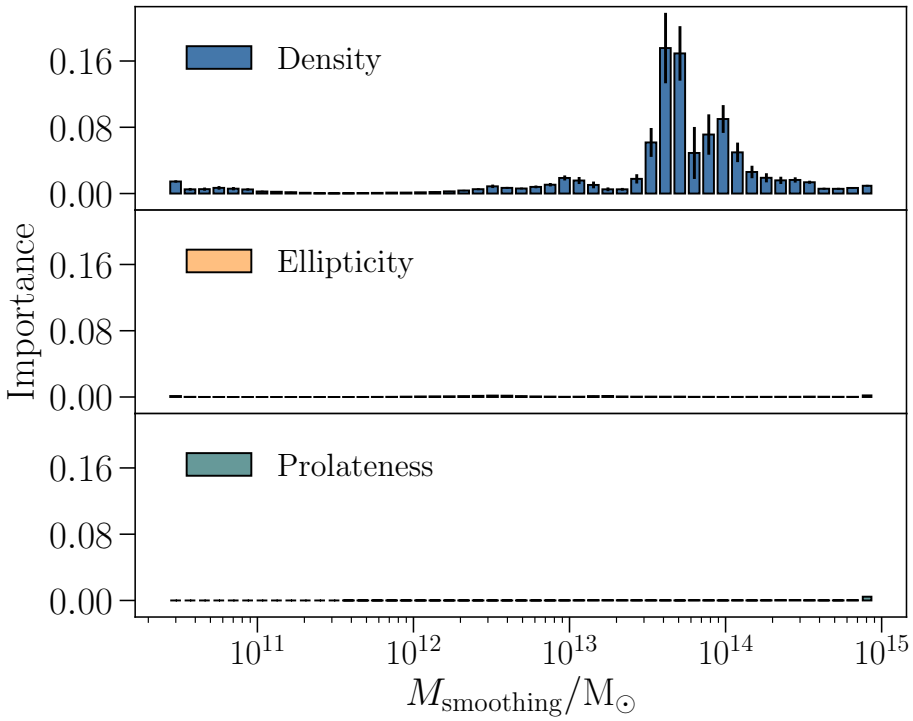

Figure 4.3: Feature importances for density (upper panel), ellipticity (middle panel) and prolateness (lower panel) as a function of the top-hat window function smoothing mass scale, when the gradient boosted trees are trained on the shear and density feature set. The ellipticity and prolateness features have very low importance scores, meaning that they are irrelevant compared to the density features during the training process of the algorithm. The density features are most relevant at high smoothing mass scales. This confirms that the shear field contains very little useful information compared to spherical overdensities.

predicted distributions and overlapping medians across the full mass range of haloes. We measure the fractional change in the bias and variance (as defined in Eq. (4.8) & (4.9)) of the distributions returned by the density+shear model relative to those of the density-only model for each mass bin; we find an average change of 8% in the variance and $< 1\%$ in the bias. The change in variance is common amongst all mass bins and is driven by changes in the overall width of the predicted distributions. We conclude that the addition of tidal shear does not provide major qualitative changes to the predicted final mass of haloes in the range $11.4 \leq \log(M/M_\odot) \leq 13.4$, thus generalizing the conclusions of Chapter 3 to regression over this mass range.

We now quantify which features contain the most relevant information on final halo masses, by calculating feature importances (see Sec 4.3.1) in the density+shear model. Fig. 4.3 shows that spherical overdensities on smoothing scales $10^{13} \leq M_{\rm smoothing}/{\rm M}_\odot \leq 10^{14}$ are most informative

for predicting the mass of haloes in the range $11.4 \leq \log(M/M_{\odot}) \leq 13.4$. The importances of the density features in the density-only model also have a peak and a spread at similar smoothing mass scales. The low importance of the shear features indicates that these have very little impact on the training process of the algorithm. This confirms that information about the tidal shear is not useful compared to that of spherical overdensities.

We now show that this result also holds when the algorithm is trained to infer the formation of haloes at higher redshifts. Fig. 4.4 shows the density and shear feature importances, for the case of training the algorithm to predict the mass of the halo to which each dark matter particle will belong at $z = 2.1$. Similar to the $z = 0$ case, the ellipticity and prolateness features have negligible importance scores, meaning that the tidal shear field contains no additional relevant information over that contained in the density features about the formation of haloes at early times. The density feature importances peak at smaller smoothing mass scales, i.e. $10^{12} \lesssim M_{\rm smoothing}/{\rm M}_{\odot} \lesssim 10^{13}$, directly reflecting the fact that larger scales are still linear at $z = 2.1$ and consequently, haloes of mass $M \gtrsim 4 \times 10^{13}$ ${\rm M}_{\odot}$ have not yet formed.

To ensure our results capture at least as much information in the features as existing approximations, we validate the $z = 0$ machine learning models against existing analytic approximations. We compare the accuracy of the machine learning predictions against those of analytic theories which also provide final halo mass predictions based on the same initial conditions information. We expect the machine learning algorithm to perform (at least) as well as analytic models. If this was not the case, it would indicate that the features contain relevant information which the algorithm fails to learn, which would in turn invalidate our conclusions. The results are shown in Appendix A.1; analytic and machine learning based models yield qualitatively comparable predictions, but with smaller scatter in the predictions of the machine learning model.

4.4.1 Dependence on radial positions

We next investigated the dependence of the predictions on the radial position of particles inside haloes. This analysis was done separately for three different mass ranges of haloes. We first sub-divided particles into three equally-spaced mass ranges based on the mass of their host halo: particles in low-mass haloes ($11.42 \leq \log(M/{\rm M}_{\odot}) < 12.08$), particles in mid-mass haloes ($12.08 \leq \log(M/{\rm M}_{\odot} < 12.75$) and particles in high-mass haloes ($12.75 \leq \log(M/{\rm M}_{\odot} \leq 13.4$). For each halo mass range, we further split the particles into three categories based on their radial position with respect to the halo's virial radius $r_{\rm vir}$: particles in the innermost region of a halo ($r/r_{\rm vir} \leq 0.1$), those

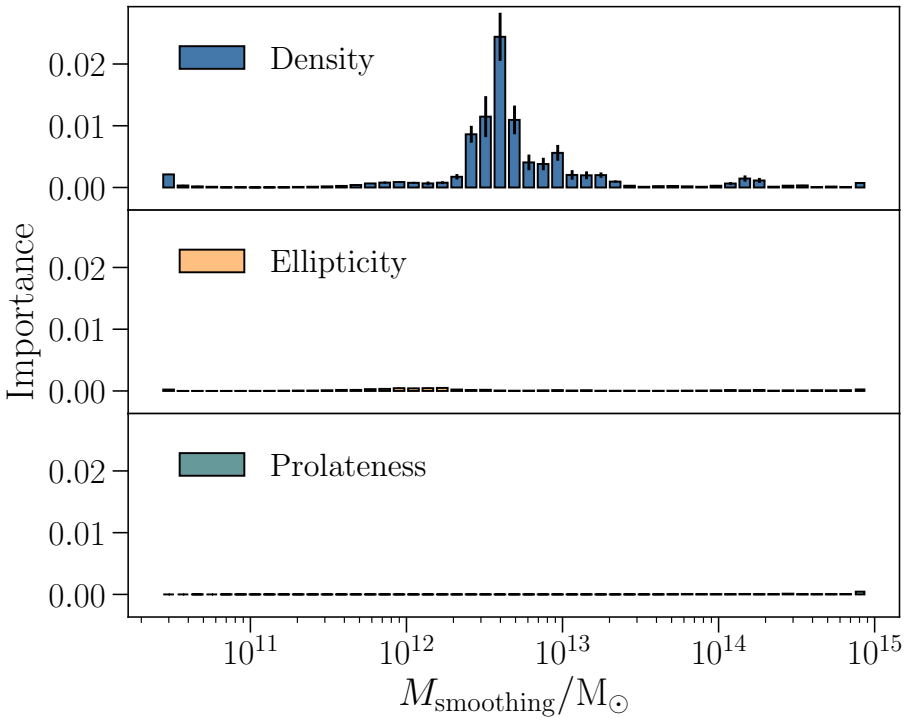

Figure 4.4: Feature importances for density (upper panel), ellipticity (middle panel) and prolateness (lower panel) as a function of the top-hat window function smoothing mass scale, for the case where the algorithm is trained to predict the mass of the halo to which each dark matter particle will belong at $z = 2.1$. Similar to the $z = 0$ case, the ellipticity and prolateness features have very little impact on the training process of the algorithm and the most relevant information is contained within the density features. The peak of the density feature importances shifts towards smaller smoothing scales, as a result of larger scales still being in the linear regime at $z = 2.1$.

in a shell of mid radial range ($0.4 \leq r/r_{\mathrm{vir}} \leq 0.6$) and those in the outskirts of haloes ($r/r_{\mathrm{vir}} > 0.8$).

Figure 4.5 shows the distributions of $\log(M_{\mathrm{predicted}}/M_{\mathrm{true}})$ values of particles in each radial category predicted by the machine learning algorithm based on the density features. The three panels show the predictions of particles in low-mass (left), mid-mass (center) and high-mass (right) haloes. For low-mass haloes, the comparison between the distributions of the three radial categories shows very little difference, indicating that the machine learning algorithm predicts the final halo mass irrespective of their final position inside the haloes. On the other hand, we find a clear improvement in the predictions for particles in the innermost regions of mid-mass and high-mass haloes. The variance of the inner particles' predictions decreases by 35% and 45% for mid-mass and high-mass haloes respectively, compared to the variance of the mid-radial particles' predictions. In high mass haloes, we also note a reduction in the bias of the distributions as one approaches the

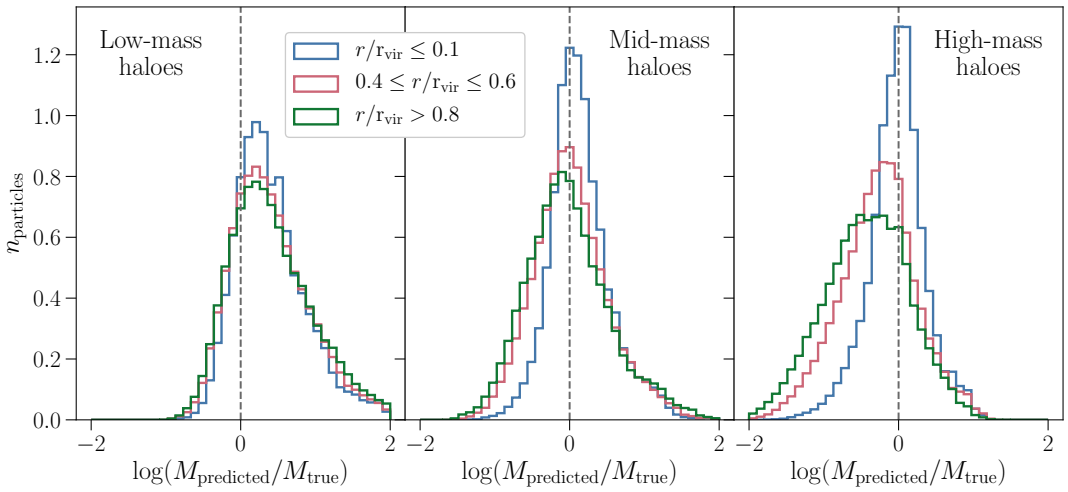

Figure 4.5: Distributions of $\log(M_{\mathrm{predicted}}/M_{\mathrm{true}})$ values for particles of different categories based on their radial position inside haloes. The panels show the distributions for particles in low-mass haloes (*left*), $11.42 \leq \log(M/\mathrm{M}_{\odot}) < 12.08$, mid-mass haloes (*center*) ,$12.08 \leq \log(M/\mathrm{M}_{\odot} < 12.75$, and high-mass (*right*) haloes, $12.75 \leq \log(M/\mathrm{M}_{\odot} \leq 13.4$. The predictions of particles in low-mass haloes are uncorrelated with the particles' radial position inside the halo. For mid-mass and high-mass haloes, particles in the innermost regions of haloes are those with highest accuracy in their predicted halo masses, compared to mid-radial and outskirts particles. The density-and-shear model produces similar distributions to those returned by the density-only model in all radius and mass bins.

haloes' central region; the medians of the $\log(M_{\mathrm{predicted}}/M_{\mathrm{true}})$ distributions are -0.0006, -0.2527 and -0.4101 for inner, mid and outer radial categories, respectively. The density and shear model produces similar distributions to those returned by the density-only model.

The correlation between the accuracy of the predictions and the radial positions of particles inside their haloes is present in high mass haloes but not within low-mass ones. One possible reason for this may be the inherent difference in their assembly history. Low-mass haloes tend to accrete most of their mass at early times, whilst more massive haloes show substantial late-time mass accumulation (Wechsler et al. 2002). As high mass haloes are thought to undergo a larger number of merger events (Fakhouri et al. 2010; Genel et al. 2009), the haloes may be characterized by a more complicated assembly history. In particular, particles in the outskirts of these haloes will be those that are particularly affected by late-time mergers, thus making it more difficult for the machine learning algorithm to infer their final halo mass based on their initial state.

4.5 A metric for machine learning model comparison

Up to this point, we have made conclusions based on visual comparisons between the predictions based on the density feature set and the density and shear one. Qualitatively, we find that the addition of tidal shear information does not yield major changes in the halo mass predictions across the whole mass range considered here. However, we require a quantitative measure of the comparison to assert whether the tidal shear contains any information that allows for a better description of halo collapse, even if minimal.

To our knowledge, there exists no metric used in machine learning regression problems suitable for judging whether one machine learning model is preferred over another. Some of the most popular metrics used to quantify the quality of the predictions are the mean absolute error, the mean squared error or the coefficient of determination (r^2). These are summary statistics which provide a measure of the magnitude of the predictive error, but have no principled statistical basis and are therefore not helpful for model comparison. As one cannot construct a likelihood function from a single generative model for making predictions, we seek a metric which is (i) based on a motivated statistic and (ii) independent from the loss function optimized by the algorithm during training.

We now describe the construction of a metric which allows us to evaluate and compare the performance of machine learning models based on different feature sets. Given a set of particles and their associated halo mass labels, one can compute the number density of particles in haloes as a function of halo mass. Although the number density of particles is directly related to the number density of haloes, the resulting halo mass function cannot be meaningfully compared to existing theoretical halo mass functions due to the small range of halo masses probed by our simulations. Therefore, we choose to work with the particle number density as it is more directly related to the machine learning predictions and to the purpose of our work. The particles' ground truth halo mass labels yield a true number density distribution, $n_{\rm true}$, and those predicted by the machine learning algorithm yield a predicted number density distribution, $n_{\rm ML}$. By comparing the two distributions, we can assess how well the machine learning approximation matches the ground truth given by the simulation. To address this question, the performance of the algorithm can be measured in terms of a difference between two distributions. In order to quantify this, we adopt the widely used Kullback-Leibler (KL) divergence (Kullback and Leibler 1951).

The KL divergence is a measure rooted in information theory of the difference between two probability distributions. In general, the KL divergence of distribution Q from P, $D_{\rm KL}(P \parallel Q)$,

describes the loss of information when Q is used to approximate the reference distribution P. This is not a symmetric function, as the information content in Q about P is not equivalent to information content in P about Q. Since we are interested in assessing how well the machine-learnt distribution describes the true distribution in the simulation, we consider the KL divergence $D_{\rm KL}(n_{\rm true} \parallel n_{\rm ML})$. If $n_{\rm true}(\log M)$ and $n_{\rm ML}(\log M)$ are continuous density distributions, the KL divergence takes the form

$$D_{\rm KL}(n_{\rm true} \parallel n_{\rm ML}) = \int_{M_{\rm min}}^{M_{\rm max}} n_{\rm true}(\log M) \ln \left[\frac{n_{\rm true}(\log M)}{n_{\rm ML}(\log M)} \right] {\rm d} \log M, \tag{4.13}$$

where $M_{\rm min}$ and $M_{\rm max}$ are given by the minimum and maximum values of $\log M$ where $n_{\rm truth}(\log M) \neq 0$. It is a non-negative quantity and takes the value $D_{\rm KL}(n_{\rm true} \parallel n_{\rm ML}) = 0$ if and only if the two distributions are identical i.e., $n_{\rm true}(\log M) = n_{\rm ML}(\log M)$.

The KL divergence yields a machine learning model comparison metric: given two models based on different input features, the difference in the KL divergences of each model's prediction from the ground truth is a quantitative measure of the difference in the amount of information contained in one feature set over the other about final halo mass. The difference in the KL divergence for the two models is computed for each of the six simulations, allowing us to quantify its statistical significance. Our choice of metric will capture some, but not all, differences between the predictions of different models.

4.5.1 Kernel density estimation

To compute the KL divergence in Eq. (4.13), $n_{\rm true}(\log M)$ and $n_{\rm ML}(\log M)$ must be in the form of continuous probability density distributions. Given the set of true and predicted mass labels of the test set particles, we can straightforwardly obtain discrete distributions for the number density of particles in haloes within bins of logarithmic mass. To then turn these into continuous ones, we adopted a smoothing procedure known as *kernel density estimation* (KDE, Rosenblatt 1956). A KDE is a non-parametric approach to estimate the probability density distribution from a discrete set of samples. Each data point is replaced with a kernel of a set width and the density estimator is given by the sum over all kernels.

For the case of the true number density, its kernel density estimate was computed from the set

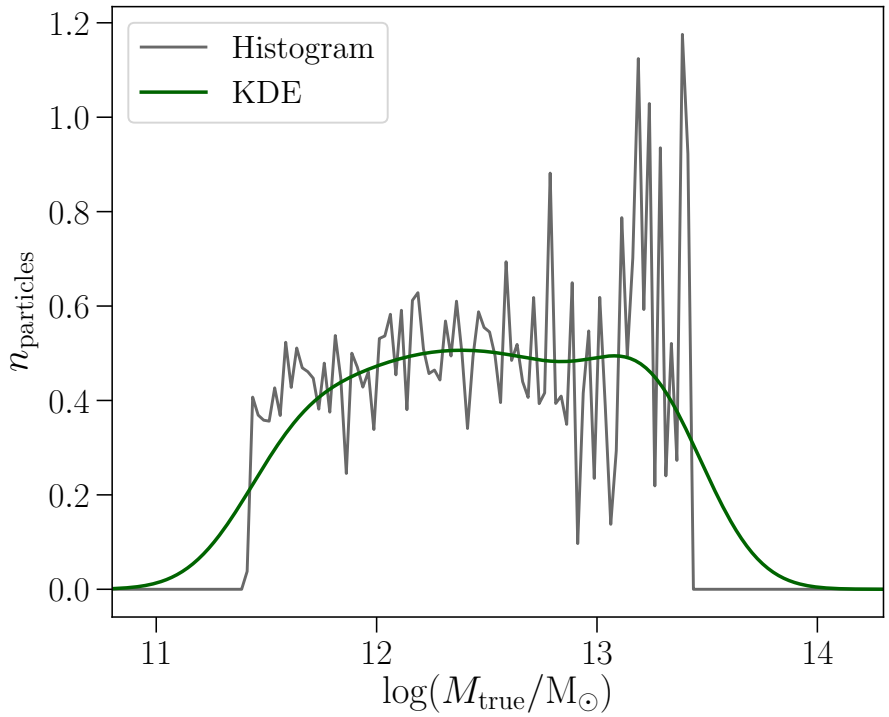

Figure 4.6: The distribution of test-set particles as a function of the logarithmic mass of the halo to which they belong at $z = 0$. The distribution is smoothed using a kernel density estimation method, where the bandwidth is optimized using cross-validation. The upper and lower limits of the binned distribution are given by $\log(M/\mathrm{M}_\odot) = 11.4$ and $\log(M/\mathrm{M}_\odot) = 13.4$, respectively.

of N ground truth logarithmic halo masses, $\{\log M^i_{\text{true}}\}^N_1$, and is given by

$$n_{\text{true}}(\log M) = \frac{1}{N}\sum_{i=1}^{N} K\left(\frac{\log M - \log M^i_{\text{true}}}{b}\right), \tag{4.14}$$

where K is the kernel, which we take to be a Gaussian of the form $K(x) \propto \exp(-x^2/2)$, and b is a smoothing parameter known as the bandwidth, which determines the width of the kernel. The bandwidth is a free parameter which strongly influences the resulting estimate. If the bandwidth is too small, the density estimate will be undersmoothed and fit too closely the small-scale structure of the simulation's distribution. If the bandwidth is too large, the density estimate will be oversmoothed meaning that it will wash out important features of the underlying structure.

We optimized the bandwidth following a 5-fold cross validation procedure, similar to the one used to optimize the machine learning hyperparameters (see Sec. 4.3.3). For a set of bandwidth values, the KDE was fitted on the simulation's true number density distribution and validated on the distribution of an independent simulation with a different initial conditions realization. To avoid

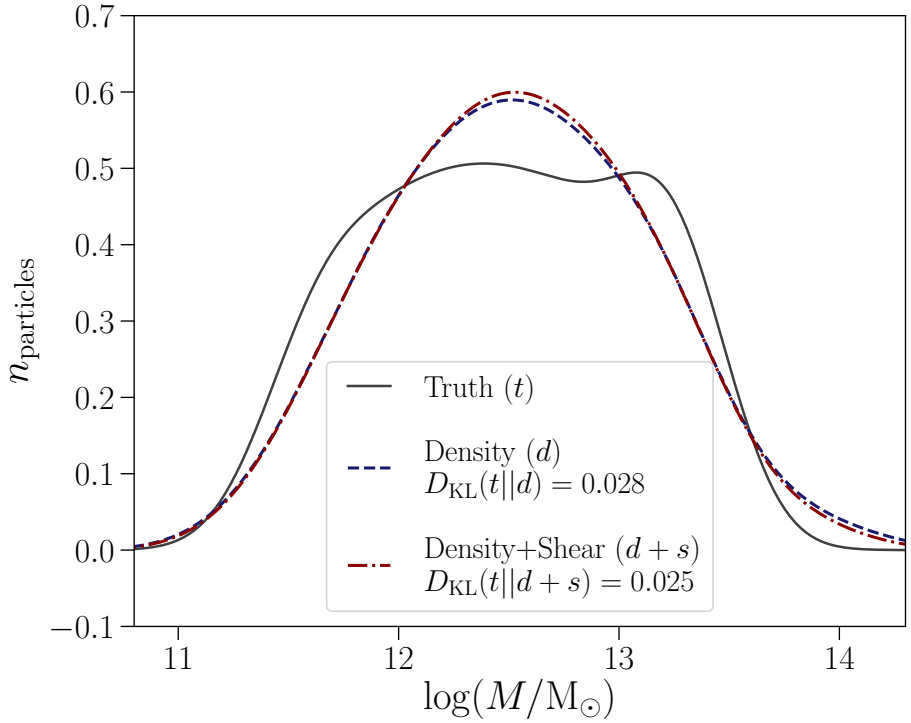

Figure 4.7: Predicted distribution of the *sim*-1 test particles as a function of logarithmic halo mass for the two machine learning models, one trained with density features and the other trained on density and shear features. The ground truth distribution is also shown for comparison. We compute the KL divergence of each model's distribution with respect to the ground truth in order to quantify and compare the model's ability to approximate the true distribution. The density and shear model yields a small improvement of 0.0029 in the KL divergence compared to the density-only model.

undersmoothing, we split the range of $\log M$ covered by the distribution into ten sub-intervals of width $\Delta \log(M/\mathrm{M}_\odot) = 0.2$ and used different mass intervals to fit and validate the KDE; every other mass bin is used for fitting and the remaining bins for validating[6]. We retained the value of the bandwidth giving the highest total log-likelihood for the validation set.

We smoothed each simulation's own ground truth number density of particles. For validation, all simulations used the ground truth distribution of *sim*-2, except for *sim*-2 which used the ground truth of *sim*-1. All six simulations returned an optimal bandwidth $b = 0.23$. The resulting kernel density estimate for *sim*-1 is shown in Fig. 4.6, together with its discrete version for comparison. We then constructed density estimates from the mass values predicted by the two machine learning models, using a KDE of the same bandwidth as for the ground truth distribution. Fig. 4.7 shows the comparison between the continuous number densities of particles in haloes based on the ground

[6]Note that this is the same binwidth we adopted in Sec. 4.3.4. This choice was made to yield at least ten mass intervals for analysis, as this is the number of bins required to carry out this bandwidth optimization procedure.

truth and the two machine learning models. Finally, we computed the KL divergence (as in Eq. 4.13) for the two machine learning models with respect to the ground truth in all six simulations.

4.5.2 Comparing KL divergences from different simulations

The final step consists of comparing the KL divergences returned by the different simulations to estimate the statistical significance of our results. To do this, we first tested the validity of comparing KL divergences across different simulations. In general, a comparison between two KL divergences has a clear meaning only if they measure differences with respect to the same reference distribution. Here, the reference distributions are different; the KL divergences we wish to compare are computed with respect to each simulation's own true number density of particles in haloes. We checked whether the ground truth distributions from different realizations are similar enough for the comparison between KL divergences to be valid. We computed $D_{\rm KL}(n_{\rm true-1} \parallel n_{\rm true-\#})$, which we denote as T for simplicity, to find the difference between each simulation's own ground truth distribution and that of *sim*-1. The values of the KL divergences are reported in the last column of Table 4.1. We find that $\overline{T}$ is at least five times smaller than any $D_{\rm KL}(n_{\rm true} \parallel n_{\rm ML})$. Therefore, the ground truth distributions are similar enough to validate the use of the KL divergence metric in the following.

4.6 Results

We present our results in Table 4.1. The first three columns show the values of $D_{\rm KL}(n_{\rm true} \parallel n_{\rm density})$, $D_{\rm KL}(n_{\rm true} \parallel n_{\rm density+shear})$ and the difference between the two, $D_{\rm KL}(n_{\rm true} \parallel n_{\rm density}) - D_{\rm KL}(n_{\rm true} \parallel n_{\rm density+shear})$ for all six simulations. We call these D, S and DS respectively, to simplify the notation. For each column X, we also compute the mean over the six realizations, $\overline{X}$, the sample standard deviation, δX, and the standard error on the mean, $\delta\overline{X} = \delta X/\sqrt{N}$, where $N = 6$ simulations.

The values of DS indicate the change in the KL divergence as we add information about the tidal shear in all six simulations. We measured the statistical significance of the deviation of $\overline{DS}$ from 0 given its standard error $\delta\overline{DS}$. We find an improvement in the KL divergence (at the 4–sigma level) provided by the addition of shear information relative to a model based on density information alone. We quantify the practical utility of such an improvement by comparing the value of $\overline{DS}$ with δD, the scatter in the density-only model. We find that the improvement provided by shear information is equivalent to a 0.5–sigma deviation from the mean KL divergence of the density-only

Table 4.1: KL divergences of a model's predicted number density of particles in haloes as a function of halo mass with respect to the ground truth distribution. Results for the density-only model (D) and density and shear model (S) of all six simulations are given in the first two numerical columns. The difference in KL divergence between the two models (DS) is shown in the third column. The algorithm was trained on each simulation independently and tested on the remaining dark matter particles in that simulation not used for training. The next three columns report the KL divergences obtained with predictions made by a machine learning algorithm trained on *sim*-1 and validated on *sim*-2. The trained algorithm is tested on *sim*-3, -4, -5, -6 and the results are shown for the density-only model (DG), density and shear model (SG) and the difference between the two (DSG). The last column shows the KL divergence of each simulation's own ground truth distribution and that of *sim*-1, $D_{\rm KL}(n_{\rm true-1} \parallel n_{\rm true-\#})$, used to validate the comparison between KL divergences of different simulations. For all columns, the last three rows show the mean, $\overline{X}$, the sample standard deviation, δX, and the standard error on the mean, $\delta\overline{X} = \delta X/\sqrt{N}$.

Sim	***D***	***S***	***DS***	***DG***	***SG***	***DSG***	***T***
1	0.0284	0.0255	0.0029	-	-	-	-
2	0.043	0.0371	0.0059	-	-	-	0.0038
3	0.0419	0.0401	0.0018	0.0597	0.0616	-0.0019	0.0055
4	0.0413	0.038	0.0032	0.0488	0.055	-0.0062	0.0045
5	0.0387	0.0286	0.0101	0.0519	0.0577	-0.0058	0.0127
6	0.0188	0.0136	0.0052	0.0361	0.0361	0	0.0027
$\overline{X}$	0.0353	0.0305	0.0049	0.0491	0.0526	-0.0035	0.0058
$\delta\overline{X}$	0.004	0.0041	0.0012	0.0049	0.0057	0.0015	0.0018
δX	0.0097	0.0101	0.003	0.0098	0.0113	0.003	0.0039

model. Therefore, we conclude that the improvement provided by the tidal shear is not large enough to yield a useful alternative model to one based on density information alone. These conclusions are consistent with the results of the feature importance analysis in Sec. 4.4.

4.7 A test of generalizability

The results presented above are valid for the case where the dark matter particles that make up the training set and the test set come from the same simulation. To test the robustness of our results, we verified the ability of the machine learning algorithm trained on one simulation to generalize to independent simulations based on different initial conditions realizations. In particular, we tested whether our main results about the significance and the utility of the improvement provided by tidal shear information still holds when generalizing to independent simulations.

We used the machine learning algorithm trained on *sim*-1 and tested it on all dark matter particles in *sim*-3, -4, -5, -6, which are independent from the training process of *sim*-1. Since the dark matter particles in *sim*-2 form the validation sets used during training, we excluded the latter from this analysis. As before, we computed the KL divergences $D_{\rm KL}(n_{\rm true} \parallel n_{\rm density})$, $D_{\rm KL}(n_{\rm true} \parallel n_{\rm density+shear})$ and the difference between the two, $D_{\rm KL}(n_{\rm true} \parallel n_{\rm density}) - D_{\rm KL}(n_{\rm true} \parallel n_{\rm density+shear})$; the values of these quantities for the four independent test simulations are reported in the fourth, fifth and sixth columns of Table 4.1. This time we call these DG, SG and DSG respectively, to distinguish them from the previous case where the test set and training set are constructed from the same simulation.

First, we tested the generalisability of each machine learning model individually. For the density feature set, the mean KL divergence computed from the independent test sets ($\overline{DG}$) is consistent (at the 2.2–sigma level) with that found when training and testing on the same simulation ($\overline{D}$), meaning that the model learnt on one simulation can indeed generalize to independent simulations. This confirms that the machine learning algorithm was able to learn the underlying physics relating the initial conditions to the final haloes. On the other hand, the model based on density and shear features shows evidence of poor generalisability, as the KL divergences $\overline{SG}$ and $\overline{S}$ are in tension at the 3.2–sigma level.

We then moved on to test the generalisability of our results regarding the improvement provided by the addition of tidal shear information. We find that the difference in the KL divergence of the two models ($\overline{DSG}$) is in significant tension (at the 4.3–sigma level) with that found when testing on the same simulation used for training ($\overline{DS}$). Moreover, as $\overline{DSG}$ was a negative value, the addition

of tidal shear information now yields a marginal loss in performance, rather than an improvement.

These discrepancies provide some evidence that the algorithm trained on density and shear features overfits the simulation during training. This naturally yields better predictions when testing the algorithm on the simulation used for training compared to testing on independent simulations. Consequently, the addition of tidal shear information yields an improvement or a loss in performance compared to the density-only model, depending on whether the algorithm is tested on the same or a different simulation to that used for training. In spite of this, the level of overfitting in the density and shear model is small; for both cases, the change in KL divergence between the two models ($\overline{DS}$, or $\overline{DSG}$) is consistent with the scatter in the density-only model (δD, or δDG).

In summary, the algorithm trained on density information has learnt the physical connection between the initial conditions and the final haloes, as it is able to generalize to independent realizations of the initial density field. On the other hand, the improvement in the KL divergence provided by the addition of tidal shear features is lost when applying the trained algorithm to independent simulations. Therefore the improvement from including shear features in the machine learning process, which was anyway small, does not imply any physical connection. This strengthens our conclusion that there is no identifiable physical information pertinent to the final halo mass in the tidal shear field.

These conclusions were made by testing the algorithm on independent realizations with fixed cosmological parameters. The parameters of the ΛCDM model are so tightly constrained from current observations (e.g. Planck Collaboration et al. 2018a), that the formation of haloes must proceed in a similar way at the mass scales investigated in our analysis. Therefore, we expect no significant change in our results when adopting simulations based on different choices of cosmological parameters. Moreover, we expect similar results for the mass range considered in this analysis for observationally-allowed cosmological models which suppress small-scale power; in such models halo abundances differ from ΛCDM only below $M \sim 10^{11}\ \mathrm{M}_\odot$.

Our results for the halo mass range $11.4 \leq \log(M/\mathrm{M}_\odot) \leq 13.4$ are also expected to hold for simulations of different box sizes or resolutions. In particular, a simulation with larger box size or higher resolution yields the possibility of extracting additional features at larger or smaller smoothing scales, respectively. Since the feature importances (Fig. 4.3) show that the most relevant information is contained within features on smoothing scales $10^{13} \leq M_{\mathrm{smoothing}}/\mathrm{M}_\odot \leq 10^{14}$, the results do not change when the simulation contains additional small- or large-scale information. Similarly, our results should hold for simulations of smaller box sizes and/or lower resolutions, as long as those scales which carry the most relevant information are resolved.

4.8 Conclusions

We have presented a generalization of the work in Chapter 3, which explored the impact of different initial linear fields on the formation of dark matter haloes above or below a single mass threshold. In this paper, we investigated a wider mass range of dark matter haloes and their sensitivity to the initial density and tidal shear fields.

We find that the tidal shear field does not contain additional information over that already contained in the linear density field about the formation of dark matter haloes in the mass range $11.4 \leq \log(M/M_{\odot}) \leq 13.4$. We quantified this using a machine learning regression framework, showing that the results are physically interpretable and generalisable to independent realizations of the initial density field. Interpretability is achieved by comparing machine learning models based on different input properties of the initial conditions; the addition of tidal shear information yields a halo collapse model whose predictions are statistically consistent with those of a model based on density information alone, according to a metric based on the Kullback-Leibler divergence. By measuring the feature importances of the different inputs during the training process of the algorithm, we can establish a complementary measure of which physical aspects contain the most information about halo collapse. This analysis confirms that our machine learning approach suggests little role for the tidal shear field in establishing final halo masses. This result holds also for the case of predicting the mass of haloes at $z = 2.1$. Generalisability is verified by applying the machine learning algorithm trained on one simulation to independent simulations based on different realizations of the initial density field. This allows us to confirm the ability of the machine learning algorithm to learn physical connections between the initial conditions and the final dark matter haloes.

In future work, we also plan to consider the relation between the initial conditions and other cosmic web structures, such as sheets, filaments and voids. The machine learning framework used in this work was carefully constructed for the purpose of studying halo formation. This involved choosing (i) suitable volume and resolution for the N-body simulations, that yield a representative set of halos for training, (ii) features motivated by existing analytic theories of halo collapse and (iii) performance metrics that reflected different science questions about halo formation specifically. All these choices must be revisited when applying our machine learning approach to other cosmic web structures. Moreover, factors which did not impact our results on halo formation may become important for other structures, as for example the choice of “ground truth” definition for voids and filaments. Therefore, although our machine learning framework can be applied to any cosmic web

structure, studying the formation of voids and filaments goes beyond the scope of this thesis and will be subject of future work.

Our work demonstrates the utility of machine learning techniques to gain physical understanding of large-scale structure formation. The strength of this approach lies in its ability to establish a physical interpretation of the machine learning results. In future work, we also plan to extend our framework to develop interpretable deep learning algorithms, aiming to learn directly from the initial density field which physical aspects are most relevant to cosmological structure formation, beyond spherical overdensities and tidal shear forces.

Acknowledgements

We thank Justin Alsing, Boris Leistedt, Michelle Lochner, Jason McEwen, Daniel Mortlock, Nikos Nikolaou, Martin Rey and Ravi Sheth for useful discussions. LLS acknowledges the hospitality of the Oskar Klein Centre, Stockholm where part of this work was completed. LLS was supported by the Science and Technology Facilities Council. HVP was partially supported by the European Research Council (ERC) under the European Community's Seventh Framework Programme (FP7/2007-2013)/ERC grant agreement number 306478- CosmicDawn, and the research project grant "Fundamental Physics from Cosmological Surveys" funded by the Swedish Research Council (VR) under Dnr 2017-04212. AP was supported by the Royal Society. This work was partially enabled by funding from the UCL Cosmoparticle Initiative.

5 A deep learning model for dark matter halo formation

5.1 Abstract

We develop a deep learning framework, based on 3D convolutional neural networks (CNNs), which maps the initial conditions density field to the final dark matter halo masses, trained on N-body simulations. We compare the performance of the deep learning algorithm to that of machine learning models whose predictions are based only on spherically-averaged overdensity features. Despite the fact that the inputs provided to the CNN contain all the information needed to fully describe the initial conditions of the Universe, the predictions returned by the CNN are consistent with those returned by machine learning algorithms trained on spherical overdensities alone. This result may suggest that the features learnt by the CNN from the initial conditions to infer halo masses at $z = 0$ resemble those of spherically-averaged overdensities. This work presents the first step towards our broader goal of utilizing machine learning for *knowledge extraction*: we aim to gain new physical understanding of halo formation by extracting information from the deep learning model regarding the underlying physics of halo collapse.

5.2 Introduction

An outstanding problem in cosmology is to understand the complex evolution of the Universe from its near-uniform early state to the present-day clustered distribution of matter. Dark matter haloes grow over time via gravitational instability, starting from small random perturbations present in the linear density field at early times. The non-linear nature of gravitational evolution makes it difficult to understand how the linear initial conditions map onto the present-day large-scale structure. Previous studies have provided a qualitative understanding of halo formation by using simple

analytic models that require restrictive assumptions about the physical processes involved, such as spherical or ellipsoidal collapse, and are implemented in the context of excursion set theories (Bond and Myers 1996; Bond et al. 1991; Doroshkevich 1970; Press and Schechter 1974).

In Chapters 3 & 4, we proposed a novel approach based on machine learning to gain new insights into physical aspects of the early universe responsible for halo collapse, without the need to introduce approximate halo collapse models. The approach consists of training a machine learning algorithm to learn the relationship between the early universe and late-time halo masses directly from numerical simulations. The learning of the algorithm is based on a set of inputs, known as *features*, describing specific physical aspects about the linear density field in the initial conditions. Our choice of inputs was motivated by existing analytic approximations of halo collapse; we provided the algorithm with spherical overdensities (motivated by spherical collapse theories) and tidal shear information (motivated by ellipsoidal collapse theories) in the local environment surrounding each dark matter particle in the initial conditions. Contrary to existing interpretations of the Sheth-Tormen ellipsoidal collapse model, we found that the addition of tidal shear information was unable to yield an improved model of halo collapse compared to a model based on density information alone (Lucie-Smith et al. 2018, 2019). This approach is limited by the need for feature extraction, a step required by most standard machine learning algorithms; in order to propose a set of informative features, we must rely on our current understanding of halo formation based on simplified and incomplete analytic approximations of halo collapse.

In this Chapter, we extend our approach to a deep learning framework based on convolutional neural networks (CNNs; Bengio 2009; LeCun et al. 2015). Convolutional neural networks are a special family of deep learning algorithms, capable of extracting meaningful and spatially-local information directly from the raw data. Therefore, we can train a CNN to learn about halo formation directly from the initial density field, without the need to manually extract features from the initial conditions. We train a CNN to predict the mass of the halo to which each dark matter particle belongs at $z = 0$. The input is given by the 3D initial density field sampled within a box, centred on the particle's initial position. As explained in detail in Chapter 2, most CNN models follow a similar architecture, made of two main components: a feature extraction part and a predictive part. The feature extraction part consists of a series of *convolutional layers*, in which the algorithm learns to extract important features from the input data. The input data is repeatedly convolved with different kernels (or, filters), each designed to detect a specific type of feature present in the input. Convolutional layers are stacked onto each other by using the output of one layer as the input to the next layer. This hierarchical structure of deep learning models enables the network to learn

complex features. The idea is that the first layers learn local low-level features, which are then combined by subsequent layers into more global, higher-level features (Le et al. 2011). The second component of the CNN is the predictive part; the features assembled in the convolutional layers are combined to return the final output via *fully-connected layers*. Fully-connected layers are made of *neurons*, each returning a single output by applying a non-linear function to a weighted sum of the inputs. The ability of CNNs to generate features makes them an attractive proposition to gain new insights into non-linear structure formation; they can learn directly from the initial conditions which physical aspects are most useful to predict final halo masses. On the other hand, presenting a machine learning algorithm with a pre-defined set of properties carrying physical meaning has the advantage of being easily interpretable. As discussed in Chapter 2, interpretability in deep learning remains a challenge and is an active area of research in the machine learning community (see e.g. Olah et al. 2018).

Although CNNs are generally applied to two-dimensional images, we employ CNNs with three-dimensional kernels that can be applied to the 3D initial density field of the N-body simulation. Applications of CNNs to three-dimensional data are very recent and mostly limited to 3D medical image segmentation in the machine learning community (e.g. Kamnitsas et al. 2015). In cosmology, 3D CNNs were first applied to N-body simulations in Ravanbakhsh et al. (2016) to estimate cosmological parameters from the 3D dark matter distribution. This work was extended to different deep learning frameworks for a similar purpose (e.g. Mathuriya et al. 2018; Pan et al. 2019) and to estimate the parameters from 3D simulated galaxy maps (Ntampaka et al. 2019). CNNs have also been employed to learn mappings which require expensive N-body simulations; these include that between the Zel'dovich-displaced and the non-linear density fields (He et al. 2019), the non-linear density field and the halo distribution (Charnock et al. 2019; Kodi Ramanah et al. 2019; Modi et al. 2018) and the dark matter and galaxy distributions (Zhang et al. 2019). Moreover, CNNs have also proved useful for classifying objects in simulations, such as filaments and walls at $z = 0$ (Aragon-Calvo 2019) or protohaloes in the initial conditions (Berger and Stein 2019).

The overall goal of our work is to use deep learning for knowledge extraction i.e., to extract information about the physics driving the formation of dark matter haloes from the deep learning results. To do this, we require a deep learning framework that allows for the interpretability of its learning; for example, in understanding the features assembled by the convolutional layers and how these map onto the final predictions. In Chapter 6, we will describe how we plan to address the problem of model interpretability by using variational auto-encoders (VAEs) to reduce the 3D initial density field into a lower-dimensional representation known as *latent variables*. The latent variables

will be used as input to a feed-forward neural network, trained to predict the mass of the final dark matter haloes. The latent variables provide us with an automatically generated set of features, containing all the relevant information about the initial conditions, which can be interpreted in relation to physical aspects of the initial density field that impact the formation of late-time haloes. In this Chapter, we present the first step towards this goal. We start with a network architecture based on CNNs alone, without any VAE component, trained to infer final halo masses starting from the 3D initial density field.

This chapter is organized as follows. We describe the method in Sec. 5.3, starting with an overview of the pipeline. We then describe how we prepare the simulation data into machine learning inputs and outputs, and outline the training and testing procedure of the neural network algorithm. We present the halo mass predictions in Sec. 5.4, in the context of a binary classification task and of a regression one. We further test the robustness of our network in a simpler setting in Sec. 5.5, where the algorithm is trained starting from the non-linear density field at $z = 0$. We draw the final conclusions in Sec. 5.6 and outline future steps of this work.

5.3 Method

We start with a brief outline of our deep learning pipeline. A more detailed description of each step is provided in the next sections (Sec. 5.3.1 to 5.3.4).

The first step in any machine learning problem involves the collection and preparation of the training data. We generated the training data from existing dark-matter-only N-body simulations (see Sec. 5.3.1). The final snapshot of the simulation was used to label each dark matter particle with its ground truth target variable, whereas the initial conditions were used to extract the machine learning inputs associated with each particle. The details of the machine learning inputs and outputs used in this work are described in Sec. 5.3.2. The dark matter particles in the simulations, each with its input and ground truth label, form the set of samples used to train, validate and test the deep learning model.

The next step is choosing the architecture of the deep learning model. This involves choosing the specific sequence of layers in the CNN and the values of the hyperparameters that determine the workings of such layers. We refer the reader to Chapter 2 for an in-depth description of the different layers in a CNN, including their purpose and their hyperparameters. In Sec. 5.3.3 we outline the choices adopted in this work. The architecture was revisited by testing the response of the CNN to changes in various hyperparameters on a validation set of dark matter particles. Finally, the choices

that returned the lowest score in the validation loss were retained in the final model.

Training a deep neural network model is the most challenging part of the pipeline. First, one specifies a loss function, which measures the error in the predictions returned by the network compared to their ground truth labels. Training involves adjusting the parameters of the CNN to minimize the loss function, yielding predictions as close as possible to their respective ground truth. The optimization of the parameters is done via backpropagation, which is simply the ordinary chain rule for partial differentiation applied to solve the gradient of the loss with respect to the parameters. The training proceeds for a number of *epochs*, each consisting of a number of forward passes, where the input passes through the network and reaches the output layer, and backward passes, in which the parameters of the network are updated to minimize the loss function evaluated for the training data. The training data is generally not provided to the CNN all at once. Instead, the training samples are split into sub-sets, called *batches*, which are forward- and backward-propagated through the network independently. Each time a batch is fed-forward through the network, the weights are updated in the backward pass according to gradient of the loss evaluated for that batch. One epoch of training is completed once all batches have been seen once by the network. At the start of a new epoch, the ordering of the batches is usually shuffled to avoid the algorithm from memorizing patterns in the batch ordering.

5.3.1 Simulations

As training data, we made use of six dark-matter-only simulations produced with `P-GADGET-3` (Springel 2005; Springel et al. 2001) and a WMAP5 ΛCDM cosmological model; the cosmological parameters are given by $\Omega_\Lambda = 0.721$, $\Omega_\mathrm{m} = 0.279$, $\Omega_\mathrm{b} = 0.045$, $\sigma_8 = 0.817$, $h = 0.701$ and $n_s = 0.96$ (Dunkley et al. 2009). The simulations are denoted as *sim-#*, where $\# \in [1, 6]$. Each simulation is based on a different realization of a Gaussian random field drawn from the initial power spectrum of density fluctuations. All simulations consist of a box of comoving size $L = 50\ h^{-1}\mathrm{Mpc}$ and $N = 256^3$ dark-matter particles evolving from $z = 99$ to $z = 0$[1]. The simulations are the same as those used in Chapters 3 & 4.

Dark matter haloes were identified at $z = 0$ using the `SUBFIND` halo finder (Springel et al. 2001), a friends-of-friends method with a linking length of 0.2, with the additional requirement that particles in a halo be gravitationally bound. We consider the entire set of bound particles that make up a halo and do not account for substructure within haloes. The resolution and volume of the

[1]We made use of the Python package `PYNBODY` (Pontzen et al. 2013) to analyse the information contained in the simulation snapshots.

simulation limit the resulting range of halo masses; the lowest-mass halo has $M = 2.6 \times 10^{10}$ $\mathrm{M}_\odot$ and the highest $M = 4.1 \times 10^{14}$ $\mathrm{M}_\odot$.

5.3.2 Machine learning inputs and outputs

To train and test the algorithm, we used the final snapshot ($z = 0$) to label each dark matter particle with its ground truth. Since our goal is to derive insight into the formation of haloes, we restricted our analysis to the subset of particles that collapse into resolved haloes at $z = 0$. The ground truth output of each dark matter particle is given by the logarithmic mass of the halo to which the particle belongs at $z = 0$. Each particle, with its logarithmic halo mass label, was then traced back to its position in the initial conditions ($z = 99$) where we extracted the inputs to the deep learning algorithm.

The input for each dark matter particle is given by the initial density field sampled in a 3-D box of comoving length $L_{\mathrm{box}} = 10\,\mathrm{Mpc}\,h^{-1}$ and resolution $N = 51^3$, centred on that particle's initial position. The resolution is ultimately limited by memory consumption; the highest resolution achieved with present-day hardware is $N = 128^3$ in Mathuriya et al. (2018). The size of the sub-box should be large enough to capture large-scale information that is relevant to the algorithm to learn the initial conditions-to-halo mass mapping. In Chapter 4, we trained a gradient boosted tree to learn about halo formation given the density field smoothed on different mass scales; we found that the algorithm was able to learn relevant information from the smoothed density field up to a scale of $M_{\mathrm{smoothing}} \sim 10^{14}\mathrm{M}_\odot$. Therefore, we chose the size of a box such that it encloses a total mass of $M \sim 10^{14}\mathrm{M}_\odot$, yielding a box length $L_{\mathrm{box}} = 10\,\mathrm{Mpc}\,h^{-1}$. The resolution was chosen such that the length of each voxel, l_{voxel}, is the same as the initial grid spacing in the simulation i.e., $l_{\mathrm{voxel}} = 0.2\,\mathrm{Mpc}\,h^{-1}$ (comoving). We assigned a value for the initial density field to each voxel in the 3-D box as follows. We took the local density estimate for each dark matter particle computed by `PYNBODY` (Pontzen et al. 2013), using the 32 nearest-neighbour particles. Since particles in the initial conditions are (to a good approximation) displaced onto a grid, we assigned to each voxel the density estimate of the particle within that voxel. The final density contrast is then given by $\delta = \rho/\overline{\rho} - 1$, where $\overline{\rho}$ is the mean matter density of the Universe.

The density field constructed in this way is a very good approximation of the raw density field realization of the simulation. The initial 3D position and velocity of the dark matter particles in the simulation are computed from the density field via the Zel'dovich approximation. Therefore, we expect the density field that we provide as input to the CNN to contain the full 6D phase space

information about the dark matter particles in the initial conditions. However, we plan to test the response of the algorithm when presented with fields other than the density in future work.

In summary, starting from the initial density contrast in a 3-D volume of comoving length $L_{\rm box} = 10\,{\rm Mpc}\,h^{-1}$ centred on a dark matter particle's initial position, the algorithm is trained to predict the mass of the halo to which that particle will belong at $z = 0$. In principle, one could use the entire initial conditions box (centred at the relevant particle's position) rather than a sub-box of $L_{\rm box} = 10\,{\rm Mpc}\,h^{-1}$; however, since we do not expect relevant information to be contained at scales larger than the size of the chosen sub-box, we gain in computational speed without loss in performance by adopting a smaller-sized box.

Finally, we calculated the mean and standard deviation of voxel values across all input boxes in the training set, and rescaled all inputs to the network in the training, validation, and test sets by these values to mean 0 and standard deviation 1 before training. A similar rescaling to mean 0 and standard deviation 1 was applied to all outputs, based on the the mean and standard deviation of output values in the training set. In general, CNNs are sensitive to the scale and the dynamic range of inputs and outputs. For example, since the parameters of the network are usually given by small numbers, CNNs do not perform well when mapping inputs with small dynamic range to outputs with large dynamic range, and vice versa. Unscaled input variables can result in large weight values which usually imply an unstable learning process, while unscaled target variables can result in large loss gradient values during backpropagation which also typically leads to unstable learning. By contrast, models based on decision trees, such as the random forest and the gradient boosted trees used in Chapters 3 $\&$ 4, do not require rescaling; they work by splitting features around the median value and are therefore insensitive to the actual scales of the input features. We applied the same rescaling to inputs and outputs in order to fix the two to the same dynamic range. The specific choice of rescaling to mean 0 and standard deviation 1 was made as it returned the best performance compared to rescaling to the ranges $[-1, 1]$ or $[0, 1]$.

5.3.3 The architecture: convolutional neural networks

We used a 3D deep CNN which takes as input the initial density field in a 3D cube centred on the dark matter particle's initial position and returns the halo mass associated to that particle. In this section, we first briefly introduce the fundamental elements of our 3D CNN model and then describe the details of the architecture adopted in this work. We refer the reader to Chapter 2 for a more general discussion on CNNs, including the role, the workings and the parameters of the different

layers as well as the principles of the training process.

Our CNN model is composed of four main ingredients:

- *Convolutional layers:* they extract meaningful features from the input data by performing convolutions between the input and several kernels. The end product of each convolution is a *feature map*, indicating the strength and location of the feature detected by that kernel (see Chapter 2 for details on convolutions in CNNs). Typically, a non-linear activation function is then applied to every feature map. The convolutions were applied to our input 3D cubes using three-dimensional kernels; the main restriction when applying CNNs to 3D data (compared to 2D data) is the increase in memory consumption. However, this can be alleviated by restricting the number of filters in the first convolutional layers. Each kernel consists of a 3D array of values, known as *weights*, which are first randomly initialized and subsequently updated during the training process of the CNN. Convolutional layers also carry a number of non-trainable hyperparameters which control the way convolutions are performed and that must be set prior to training. These include the size of the kernels, which is typically set to be small (e.g. $3 \times 3 \times 3$ or $5 \times 5 \times 5$) in order to detect low-level local features across different regions of the input volume. Low-level local features are then combined into high-level global features by adding several convolutional layers to the model. Additional hyperparameters are: the number of kernels, which dictates the number of individual features detected by the CNN in a single convolutional layer, the *stride* i.e., the number of pixels by which to slide the kernel across the input when performing the convolution, and the amount of *zero-padding* i.e., whether to pad the input boxes with zeros around the borders so that the kernels can be centred on elements at the edge of the box.

- *Batch-normalization layers:* they normalize the inputs of each batch by (i) subtracting the batch mean and dividing by the batch standard deviation and (ii) rescaling and shifting the normalized values by two parameters γ and β, which are learnt during training. Batch-normalization layers do not contain hyperparameters.

- *Pooling layers:* they decrease the resolution of the feature maps by taking the average (average-pooling) or the maximum value (max-pooling) in small regions of the feature maps. The size of the pooling region is the only hyperparameter in this layer, which does not contain trainable parameters.

- *Fully-connected layers:* they connect every neuron in one layer to every neuron in adjacent

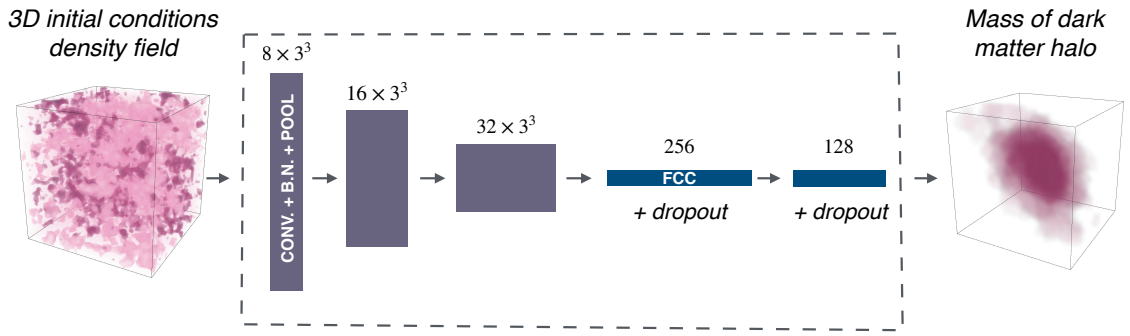

Figure 5.1: The deep learning architecture adopted in this work. The input is given by the initial density field in a 3D cube centred on a dark matter particle's initial position. The purple layer represents a convolutional layer, followed by batch-normalization (B.N.), a leaky ReLU non-linear activation function and a layer of average pooling. Above each purple step are shown the number of kernels $\times$ the size of the kernel. The blue layers are fully-connected layers with 20% dropout. Above each blue step are shown the number of neurons in each fully-connected layer. The output is given by the mass of the dark matter halo to which the dark matter particle will belong at $z = 0$.

layers. Each neuron follows $y = \sigma(\mathbf{w} * \mathbf{x} + b)$, where $\mathbf{x}$ are the inputs, y is the output, σ is the non-linear activation function and $\mathbf{w}$, b are trainable parameters known as weights and biases. To reduce the likelihood of overfitting in these layers, it is common to adopt a regularization technique known as *dropout*, where a set of randomly drawn neurons are ignored (i.e. not updated) at each epoch of training.

Our architecture is illustrated in Fig. 5.1. It consists of 3 convolutional layers, each followed by batch-normalization and pooling layers, and 3 fully-connected layers. The convolutions were performed with 8, 16, and 32 kernels for the first, second and third convolutional layer respectively, with a stride of 1 in all layers and no zero-padding. In principle, the choice of no zero-padding implies throwing away information at the edges of the box, as the convolutional kernels are never centred on those voxels. This could in principle imply that the algorithm is not learning the large-scale information. However, we tested that including zero-padding does not result in any improvement in the loss score of the validation set. This may therefore suggest that large-scale information is nevertheless captured by the algorithm in the way the feature maps of previous layers are combined in subsequent layers. The initial weights of the kernels were drawn from a Gaussian distribution of mean 0 and standard deviation 0.05, except that values more than

two standard deviations from the mean were discarded and redrawn. This is the recommended initializer for neural network weights and filters. The filters have size $3 \times 3 \times 3$ in all convolutional layers, meaning that the first layer learns features on scales of $0.6\,\mathrm{Mpc}\,h^{-1}$. As more convolutional layers are stacked on top of each other, the algorithm becomes sensitive to features at increasing scales. In this way, both local and global information are able to propagate through the network. The batch-normalization layers are placed immediately after the convolutional layers, in order to normalize the feature maps within each batch, before applying the non-linear activation function. We used a leaky rectified linear unit (LeakyReLU; Nair and Hinton 2010) activation applied to each value in the (normalized) feature maps, defined as

$$f(x) = \begin{cases} x & \text{for } x \geq 0, \\ \alpha \times x & \text{for } x < 0, \end{cases} \tag{5.1}$$

with $\alpha = 0.03$. A leaky ReLU activation, with α of order 10^{-2}, is a common choice that has proved successful in many deep learning applications (see e.g. Nwankpa et al. 2018). The feature maps are then fed to average-pooling layers, which reduce their dimensionality by taking the average of $2 \times 2 \times 2$ non-overlapping regions of the feature maps.

After the third loop of convolutional, batch-normalization and pooling layers, the output is flattened into a one-dimensional vector and fed to a series of 3 fully-connected layers, each made of 256 and 128 and 1 neuron, respectively. The non-linear activation function of the first two layers is the same ReLU activation (Eq. (5.1)) as that used in the convolutional layers, whereas the last layer has a linear activation in order for the output to represent halo mass. The weights and biases were initialized using the same truncated Gaussian distribution used for the filters of the convolutional layers. The dropout, in which 20% of neurons in the fully-connected layers are ignored during training, is adopted in both the first two layers.

We caution that we have not explored a full grid of hyperparameters for model optimization. The final architecture described in this section was that returning the best performance i.e., the lowest loss score on the validation set after convergence, amongst many, but not all, alternative models with different choices of architecture-specific and layer-specific hyperparameters. For example, we tested whether adding complexity to the model could improve the CNN's performance, but found that the model overfitted the training data as we added more convolutional layers and/or fully-connected layers to the architecture. We investigated the change in the validation loss in response to the following modifications: removing the batch-normalization layers; varying the amount of dropout

to values of 0, 0.1 and 0.5; adding one or two convolutional layers and/or fully-connected layers; doubling the number of filters at each convolutional layer; adding zero-padding; changing the convolutional kernel size to 4^3 or 2^3; changing the weight initializers to Gaussian distribution of mean 0 and standard deviation 0.1. In all cases, we found that the final loss score either increased or showed no change compared to that of the architecture retained in this work. The largest number of trials were spent on finding the right balance between batch size, learning rate and number of epochs. We tested initial learning rate values of 10^{-2}, 10^{-3} and 10^{-4}, together with batch sizes of 40, 80 and 160; we found that the algorithm converged with the least number of epochs (80) with an initial learning rate of 10^{-4} and a batch size of 80. Further hyperparameter exploration, including changes to the optimizer, the addition of skip-connections and other variations in the convolutional and fully connected layers, are part of ongoing work.

5.3.4 Training the deep learning algorithms

The algorithm is trained on $80{,}000$ particles consisting of $20,000$ dark matter particles from each of four simulations based on different initial conditions realizations. As we will show in more detail in Sec. 5.4.1, we found no improvement in the performance of the algorithm as we added to the training set an additional $20{,}000$ particles from another independent simulation, implying that four simulations were sufficient to yield a training set representative of the initial conditions-to-haloes mapping. The training set was sub-divided into 1000 batches, each made of 80 dark matter particles. In Sec. 5.4.1, we will test the perfomance of the CNN against different approaches to sub-divide the particles in the training set into batches of 80 particles each.

Training was done using the `AMSGrad` optimizer (J. Reddi et al. 2018), a variant of the widely used `Adam` optimizer (Kingma and Ba 2014), with an initial learning rate of 0.0001 which is exponentially decreased with number of epochs. The number of trained parameters in the network is $\sim 500,000$. The loss function was chosen to be the mean squared error. The network was considered to have converged and training was stopped once the validation error did not improve for ten consecutive epochs. Networks were trained on a single NVIDIA V100 GPU, using Keras with a TensorFlow backend, for ~ 10 hours and 80 epochs.

Validation was performed on the dark matter particles from one independent simulation based on a different realization of the initial density field to those used for training. Although the validation set does not directly enter the training process of the algorithm, it is indirectly used to test the response of the algorithm to changes in the architecture. The performance of the algorithm was

tested on dark matter particles from a different independent simulation. Testing on independent realizations ensures that the algorithm is not overfitting patterns specific to the simulations used for training, but rather physical connections between the initial conditions and the final haloes which are generalizable to any realization of the initial density field.

5.4 Halo mass predictions from the initial density field

In Sec. 5.3, we described the main steps of our deep learning pipeline by which particles in the initial conditions are mapped onto the final mass of the halo to which they belong at $z = 0$. Before implementing this, we tested the pipeline on a simpler setting of a binary classification problem, similar to that of Chapter 3. A convolutional neural network is trained to predict whether a given dark matter particle will later belong to a halo above or below a single mass threshold. In Sec. 5.4.1, we first outline the small changes in the data preparation and the model architecture between the binary classification and the regression setups. We then describe the validation process and how we test the robustness of the algorithm against different ways of splitting the training data into batches. We then present our results and compare them to those from Chapter 3. In Sec. 5.4.2, we finally apply the deep learning pipeline to the regression problem; the performance of the CNN is compared to that of the gradient boosted trees adopted in Chapter 4.

5.4.1 Binary classification

Training: data preparation $\&$ model architecture

We started by testing our deep learning model on the simplest setting of a binary classification problem. We split the dark matter particles in the simulations into two classes depending on whether they belong to haloes above or below a given mass threshold at $z = 0$. All dark matter particles in haloes with $M_{\rm halo} \geq 2 \times 10^{12}\,{\rm M}_{\odot}$ belong to one class, whereas those in haloes with $M_{\rm halo} < 2 \times 10^{12}\,{\rm M}_{\odot}$ belong to another class. The mass threshold $M_{\rm th} = 2 \times 10^{12}\,{\rm M}_{\odot}$ lies in the middle of the halo mass range probed by the simulation and is similar to that used in Chapter 3. The inputs to the deep learning algorithm were kept the same for the binary classification and the regression setups; the CNN was trained to infer the final class of a particle based on the initial density contrast sampled in a 3D box of comoving length $L_{\rm box} = 10\,{\rm Mpc}\,h^{-1}$ centred on the particle's initial position. The architecture of the binary classification model differs from that of the regression problem (described in Sec. 5.3.3) in two respects. The first is the loss function, which in the case of

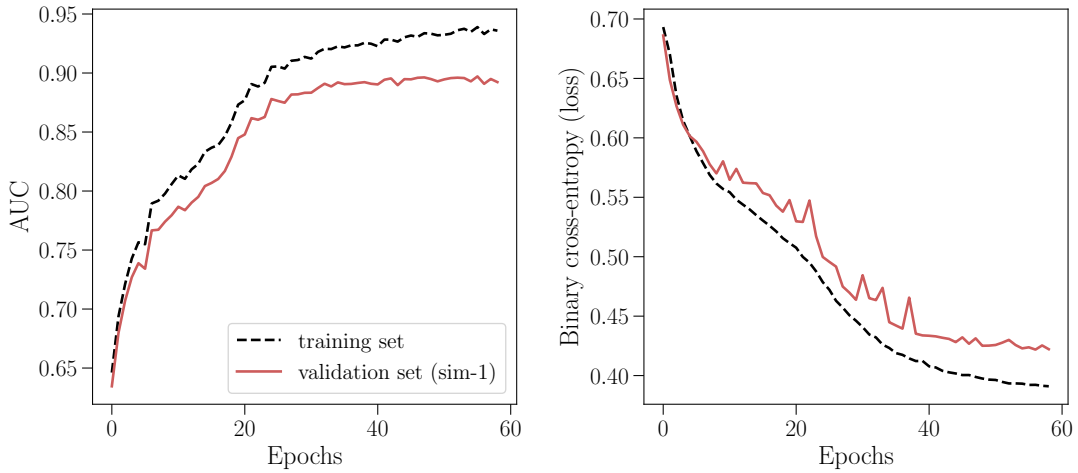

Figure 5.2: The evolution of the AUC score (*left panel*) and the loss function (*right panel*) as a function of epoch, for the training set and the validation set. The algorithm converges after 60 epochs, since the validation scores of both metrics show no improvement in the last 10 epochs. Deeper architectures and/or changes in the training procedure of the CNN show no improvement in either metrics.

binary classification is the cross-entropy loss defined as $\mathcal{L} = -(y \log p + (1-y)\log(1-p))$, where y is the class label and takes values $y \in \{0, 1\}$ and p is the predicted probability of belonging to class $y = 1$. The second is the non-linear activation function of the last fully-connected layer, which is a sigmoid function instead of a linear one. Since the output of a sigmoid function is restricted between 0 and 1, a sigmoid activation function is a common choice for binary classification where the outputs represent probabilities. The training and testing data were prepared as explained in Sec. 5.3.4.

Validation

We used the loss function and the area under the Receiver Operating Characteristic (ROC) curve, or AUC, as metrics to evaluate the performance of the algorithm. A ROC curve compares the true positive rate i.e., the fraction of correctly classified positives, to the false positive rate i.e., the fraction of negatives which have been incorrectly classified as positives, as a function of the probability threshold separating the positive and negative classes. The area under the ROC curve is a useful quantity to evaluate and compare the performance of classifiers. Fig. 5.2 shows the evolution of the AUC score (left panel) and that of the loss function (right panel) as a function of epoch, for the training set and the validation set. The algorithm converges after 60 epochs, since

the validation scores of both metrics show no improvement in the last 10 epochs.

A test of robustness: different approaches for splitting data into batches

Our training data is composed of dark matter particles from five simulations based on different initial conditions realizations. As explained in Sec. 5.3, the data is typically split into batches fed to the network one at a time, and each time the CNN updates its parameters according to the samples in that batch. As there is no unique way to sub-divide particles into batches, we investigated the response of the AUC score to three different approaches in the way the training particles are split into batches:

- *Mixed sims*: the entire set of training dark matter particles was randomly sub-divided into batches, independent of which simulation the particles come from. These batches were then used to train the CNN at every epoch, until convergence was reached.
- *Sequential 4 sims*: the algorithm was trained on particles from different simulations at different epochs. For each simulation, we took the dark matter particles belonging to that simulation, randomly sub-divided them into 250 batches and used those batches to train the algorithm for three epochs. The choice of three epochs per simulation was made empirically as it returned the highest AUC score when compared to a choice of one or five epochs. In the next three epochs, the same procedure was repeated for a different simulation, and so on. We used four simulations, meaning that the algorithm was trained on all particles in the training set by the end of 12 epochs. The algorithm was trained for as many epochs as required for convergence, by re-starting the training process from the first simulation at the end of every 12 epochs.
- *Sequential 5 sims*: this is identical to the sequential 4 sims approach, except that we used five, rather than four, simulations to train the CNN. This was useful to test whether the choice of four independent realizations in the training data is sufficient for the algorithm to generalize to independent simulations. Here, the algorithm was trained on all particles in the training set over 15 epochs (3 epochs $\times$ 5 simulations), rather than 12 as for the sequential 4 sims approach.

The sequential approaches were motivated by the idea that by training the algorithm on different simulations at different epochs, the algorithm will correct for the simulation-specific information (i.e. the non-physical information) learnt in the previous epochs and retain only the information

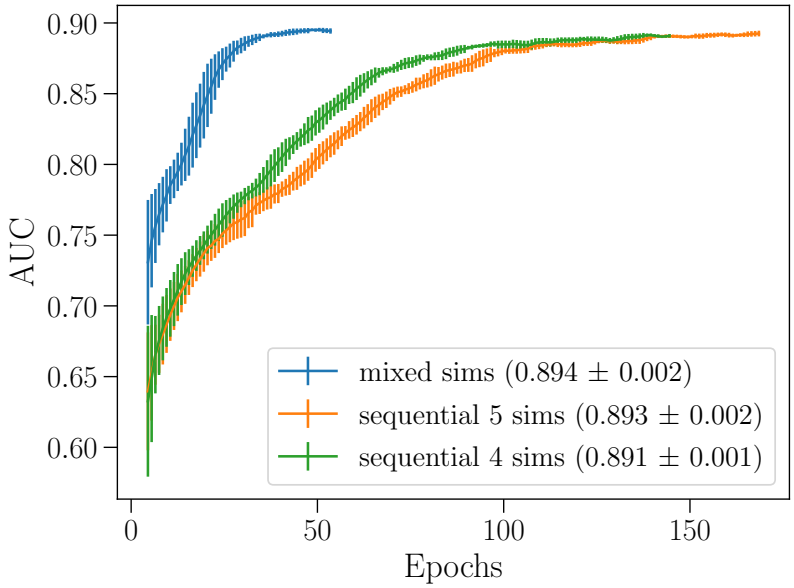

Figure 5.3: Moving average and standard deviation of the AUC score for the validation set, computed in intervals of 10 epochs for three training methods; the two sequential methods using four and five simulations respectively, and the mixed approach. The final AUCs are consistent, with the mixed approach yielding faster convergence.

that is in common amongst the different realizations. We hypothesized that this may yield better generalization to independent simulations and an improved performance.

Figure 5.3 compares the AUC as a function of epoch for the three training approaches; the two sequential methods using four and five simulations, and the mixed sims approach. We show the moving average of the AUC score over 10 epochs with its standard deviation, for a test set made of dark matter particles from a different simulation to those used in any of the training methods. The training was stopped once no significant improvement in the AUC score was found in the last 20 epochs; in all three cases, the change in the AUC (of order 10^{-4}) is smaller than the AUC's statistical error (of order 10^{-3}). The final AUCs from the three training methods are consistent; the algorithm is able to learn the relevant information contained in the training data, independent of how this information is provided during training. The mixed sims approach converges faster than the sequential approaches; this is because in the mixed approach the algorithm learns from the entire training set after a single epoch, whereas the sequential approaches take 12, or 15, epochs before the algorithm has seen the entire training data at least once.

Results

The final AUC of the test set returned by the best performing deep learning model is given by AUC$= 0.894$. We compared the performance of the deep learning model to that of the random forest model used in Chapter 3 for the same binary classification task. The fundamental difference between the deep learning (DL) and the random forest (RF) models is given by the inputs used to train the machine learning algorithms. The deep learning model learns directly from the initial density field, which fully determines the initial conditions of the universe, whereas the random forest learns from inputs which describe only specific aspects about the density field in the local environment of the dark matter particles. We find that the AUC of the deep learning model is consistent with that returned by the random forest model trained on spherical overdensities, AUC$= 0.876 \pm 0.034$. The errorbar in the AUC of the random forest comes from training the algorithm on ten training sets, each made of different subsets of randomly-chosen dark matter particles. In other words, if we provide the algorithm with the initial density field, containing all the information needed to describe the initial conditions, we recover halo mass predictions that are consistent with those of a model based on information about spherical overdensities alone. This may suggest that the CNN is extracting features which are similar to spherically-averaged overdensities, and that these saturate the most relevant information in the initial conditions about final halo masses. To verify this hypothesis, we require tools that allow us to interpret the features learnt by the deep learning model and use these to extract knowledge about the physics of the early universe relevant to halo formation. In particular, this will allow us to test whether or not the information learnt by the deep learning model coincides with that of spherical overdensities.

5.4.2 Regression

We then moved forward to applying our deep learning framework to a regression problem, where the algorithm is trained to predict the mass of the halo to which that particle will belong at $z = 0$. We followed the training procedure outlined in Sec. 5.3 and tested the performance of the algorithm on *sim*-1, the remaining independent simulation not used for training or validation.

Figure 5.4 shows the predictions made by the CNN compared to the true halo masses of the test set particles. The predictions are shown as violin plots i.e., distributions obtained from the predicted halo masses of particles within bins defined by their true logarithmic halo mass. The dots represent the medians of the predicted distributions as a function of the medians in each true mass interval. We compare the distributions resulting from the CNN with those from a gradient

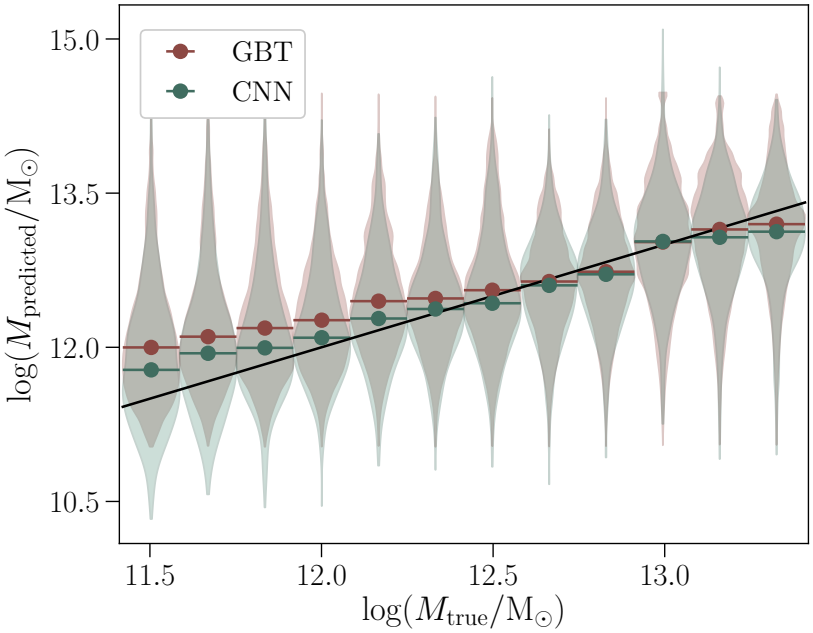

Figure 5.4: Halo mass predictions returned by a CNN trained on the initial conditions density field surrounding each dark matter particle's initial position. The predictions are shown as violin plots i.e., distributions (and their medians) of predicted halo masses of particles within evenly-spaced bins of true logarithmic halo mass. The distributions returned by the CNN are compared to those returned by a GBT, trained on spherical overdensities only. Despite the additional information contained in the inputs to the CNN, the algorithms return similar halo mass predictions with a marginal improvement in the bias of the CNN predicted distributions for low-mass haloes.

boosted tree (GBT) trained on the same regression task, similar to that adopted in Chapter 4. As for the binary classification case, the fundamental difference between the CNN and the GBT lies in the inputs we provide to the algorithms; the former learns from the initial density field, containing all the information required to describe the initial conditions, whilst the latter learns from a more limited set of information describing specific physical aspects about the density field. We find that the CNN returns qualitatively similar predictions to those from the GBT in the mass range $11.4 \leq \log(M/{\rm M}_\odot) \leq 13.4$, for the same set of test particles. The CNN provides a marginal improvement in the bias present in the predicted distributions for particles in lower-mass haloes, as shown by the medians of the CNN predicted distributions being increasingly closer to the $y = x$ line compared to the GBT medians. On the other hand, we find no significant difference in the variances of the distributions returned by the two algorithms. Overall, the fact that the CNN learns directly from the initial conditions field does not yield any substantial improvement in the final halo mass predictions, compared to training an algorithm on spherical overdensities alone. This conclusion is

consistent with that found from the binary classification case. Future work on the interpretation of the features learnt by the CNN will provide insight into whether or not there exists information beyond spherical overdensities that is useful to describe halo collapse.

5.5 A comparison with low-redshift inputs

The surprising result of the consistency between the halo mass predictions returned by the CNN and those returned by the RF motivated us to perform additional tests of the robustness of the CNN architecture adopted in this work. Choosing the exact network architecture requires extensive numerical experimentation. The large number of hyperparameters involved in convolutional neural networks makes a full optimization search, for example based on grid-search cross-validation, infeasible. As mentioned in Sec. 5.3.3, we tested the impact of changes in a variety of architecture-specific and layer-specific hyperparameters and retained the model yielding the best performance. In this section, we present additional tests to verify whether or not the CNN architecture we adopt is suited for our problem of learning final halo mass starting from the 3D density field. To do this, we tested the performance of the model in simpler scenarios where we can compare the predictions of the CNN against our expectations.

We trained the CNN to learn the mapping between the non-linear density field at $z = 0$ and the mass of the resulting haloes. This mapping is effectively given by an algorithm which first identifies the boundary of a halo based on a fixed density threshold, similar to a friends-of-friends algorithm, and then computes the mass enclosed within such halo. As this is a much simpler mapping than that between the initial conditions density field and the final halo masses, we expect the CNN to return near-perfect predictions. Similar to the $z = 99$ case, we sampled the non-linear density field at $z = 0$ in a 3D box centred at each particle's position. We revisited our choices of box size and resolution of the 3D box, as the scales of interest at $z = 0$ naturally differ from those in the initial conditions. We fixed the resolution to that used for the $z = 99$ case, $N = 51^3$, and chose a box size of $L = 1.5\,\mathrm{Mpc}\,h^{-1}$, which approximately corresponds to the virial radius of a halo with mass $M = 10^{14}\,\mathrm{M}_\odot$. These choices resulted in a voxel length $l_{\mathrm{voxel}} \sim 30\,\mathrm{kpc}\,h^{-1}$, which is approximately equivalent to half the virial radius of a $M = 10^{10}\,\mathrm{M}_\odot$ halo. Given that the box captures the virial radius of the largest and smallest haloes probed by our simulations, we expect the input boxes to contain the information required by the algorithm to learn the density field-to-haloes mapping. The training and testing of the deep learning algorithm was performed following the same procedure used for the initial conditions-to-haloes mapping in Sec. 5.4.2.

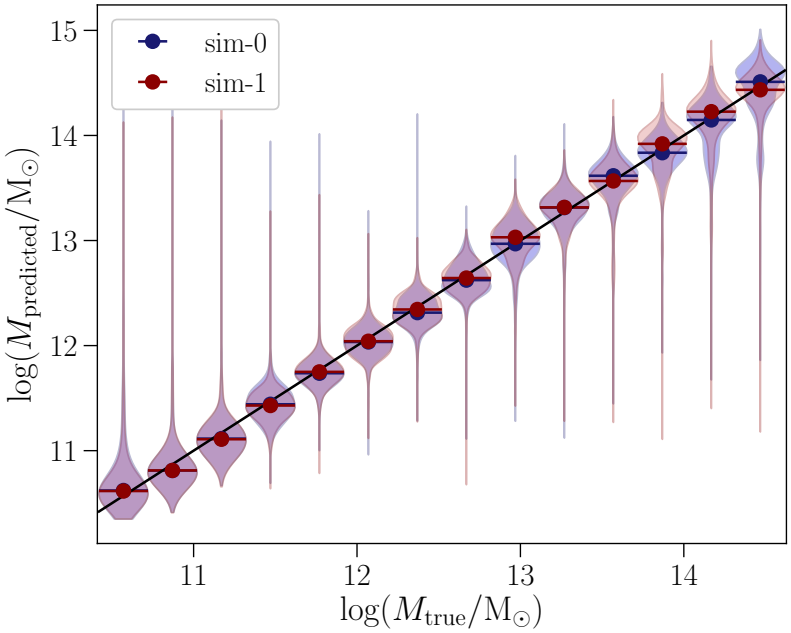

Figure 5.5: Halo mass predictions returned by a CNN trained on the non-linear density field at $z = 0$. The predictions are shown in the form of violin plots i.e., distributions (and their medians) of predicted halo masses of particles within evenly-spaced bins of true logarithmic halo mass. The distributions are shown for two independent simulations from those used for training. For both simulations, the predictions are in good agreement with their respective ground truth halo masses, yielding a Pearson correlation coefficient $r = 0.97$. However, the tails of the distributions indicate a small degree of inaccuracy in the predictions. See the text for possible origins for these tails.

Figure 5.5 shows the halo mass predictions when using the $z = 0$ non-linear density field as input to the CNN. The predictions are shown for two simulations based on different initial conditions realizations to those of the simulations used for training. They are illustrated in the form of violin plots, showing the distributions of predicted halo masses in bins of true mass. The predictions show good agreement with the true halo mass labels in both simulations, yielding in both cases a Pearson correlation coefficient $r = 0.97$, where $r = 1$ implies an exact linear relationship. Therefore, the CNN is able to make use of the information contained in the inputs and return reliable predictions. However, we note the presence of tails in the predicted halo mass distributions; these may be due to the loss of information arising from the 51^3 voxelization of the density field needed to construct the inputs to the deep learning algorithm. By contrast, the halo finder algorithm, which defines the ground truth of the neural network, works at the particle-level resolution (the highest possible resolution). This hypothesis could be verified by testing whether the model yields better predictions when trained on the initial density field sampled at a higher resolution than 51^3. Alternatively, one

could run an algorithm, similar to that of the halo finder, but working at voxel-level resolution and test whether the resulting halo mass estimates match those predicted by the CNN. Another possibility is that the tails in the distributions come from the predictions for dark matter particles which have specific properties that make it harder for the algorithm to return accurate predictions; for example, if they live near the boundary of the haloes. Further investigations regarding the origin of these tails are part of ongoing work.

A natural extension of this work would be to test the performance of the deep learning model for predicting $z = 0$ masses from the density field at intermediate redshifts, where smaller scales have already entered the non-linear regime whilst larger scales are still in the linear regime. However, this carries further complications regarding the choice of voxelization of the input boxes. The fixed choice of $N = 51^3$ resolution sets a trade-off between the large linear scales and small non-linear scales which can be probed by the box. Although an increased choice of resolution is possible, this goes at the expense of changes in the CNN architecture and larger computational costs, thus making the interpretation of the results of such tests more difficult.

5.6 Conclusions

We have presented a generalization of our machine learning framework to deep learning algorithms, capable of learning final halo masses directly from the linear density field in the initial conditions of an N-body simulation. The aim of our approach is ultimately to use deep learning for knowledge extraction; that is, we would like to learn about physical aspects of the early universe which impact the formation of late-time haloes using the results of deep learning, without the need to featurize the initial conditions as in Chapters 3 & 4.

In this work, we presented the first step towards this goal. We trained a convolutional neural network to learn the mapping between dark matter particles in the initial conditions and the mass of the halo to which the particles belong at $z = 0$. The advantage of deep learning algorithms is that they do not require feature extraction; the algorithm learns directly from the initial density field, sampled in a box centred at each dark matter particle's initial position. We compared the performance of the CNN to that of the machine learning models adopted in Chapters 3 $\&$ 4. The fundamental difference between the CNN and the machine learning models used in our previous work lies in the inputs which we provide to the algorithms. The inputs to the CNN model contain all the information necessary to fully describe the initial conditions of the universe, whereas the models in Chapters 3 $\&$ 4 learn from hand-crafted features describing only specific aspects of the

linear density field. We find that the halo mass predictions returned by the CNN are consistent with those returned by the algorithms trained on spherical overdensities alone.

One interpretation of this result is that spherical overdensities capture most of the information relevant to halo formation within the initial conditions. On the other hand, it may be that further optimization of hyperparameters of the CNN will enable improved predictions compared to those of the GBT. This may reveal additional features beyond spherical overdensities that the algorithm finds useful to predict final halo masses. To verify these hypotheses, we require tools to interpret the features learnt by the deep learning model. This will allow us to establish whether the information gleaned by the deep learning model correlates with spherical overdensities or if there exist additional physical information relevant to predict halo collapse. In the next Chapter, we outline the next steps towards building an interpretable deep learning framework for knowledge extraction: we plan to develop a network architecture that exploits the synergies between convolutional neural networks and variational auto-encoders. Future work also involves providing the deep learning model with multiple linear fields, such as the linear density field and the linear tidal shear field; this is straightforwardly achievable by exploiting the existing multiple 'channels' infrastructure already established for RGB channels in images. When an RGB image is used as input to a CNN, the input is a $N \times N \times 3$ array of pixels, where the 3 refers to the R,G and B values, which is convolved with a filter of the same depth as the input, an $M \times M \times 3$ array, where M is the size of the filter. The output of the convolutional layer is then given by a 2D feature map of depth 1, similar to the single channel case. In our context, multiple linear fields can be used as input to the CNN simply by replacing the 3D filters in the first convolutional layer with 4D filters, where the size of the fourth dimension is determined by the number of linear fields provided as input.

Our framework differs from other applications of 3D CNNs to cosmological simulations in two regards. First, our CNN model returns particle-specific predictions. This yields a halo collapse model that can describe the non-linear evolution of the density field from any initial location in the simulation. By contrast, prior work often trains 3D CNNs to infer simulation-specific quantities, such as cosmological parameters (Ntampaka et al. 2019; Pan et al. 2019; Ravanbakhsh et al. 2016), the abundance of galaxies at $z = 0$ for a given voxelization of the simulation (Zhang et al. 2019) or the $z = 0$ displacement field across the entire simulation (He et al. 2019). Predicting simulation-specific quantities requires a large number of training simulations since each simulation represents one single training example; in our work, since the training data consist of dark matter particles, we only require a few cosmological simulations, as each contains millions of particles. Two examples close to our work are that of Kodi Ramanah et al. (2019), where a Wasserstein generative adversarial

network is used to predict the halo count distribution starting from the non-linear density field, and that of Charnock et al. (2019), which uses a neural bias model to determine the halo mass distribution from the non-linear density field. These applications differ from our work not only in their implementations, but also in their aim to construct models that can be used as fast alternatives to expensive computational simulations. The second difference is in the evaluation step. We evaluate the performance of the model using localised particle-based metrics, that directly test how well the outputs of the CNN match their respective ground truth. On the other hand, existing works often evaluate the performance of their model on global summary statistics such as two-point or three-point correlation functions (e.g. Kodi Ramanah et al. 2019; Mathuriya et al. 2018; Ntampaka et al. 2019; Ravanbakhsh et al. 2016; Zhang et al. 2019). Although these are observables commonly used in cosmology, they only contain limited information about the performance of the deep learning algorithm and may therefore wash out informative aspects about the model's predictions or hide limitations of the model.

In the next Chapter, we will outline our plan for extending the current deep learning framework to one based on variational auto-encoders. This will allow us to interpret the features of the CNN by establishing whether the information extracted by the deep learning model correlates with spherical overdensities. Future work will focus on making use of this framework for knowledge extraction, to gain new physical understanding about the connection between the initial conditions and the final haloes from the learning of the algorithm itself.

Acknowledgements

LLS thanks Joao Caldeira, Andres Felipe Alba Hernandez, Michael Maire, Manuel Valentin and Samuel Witte for useful discussions. LLS acknowledges the hospitality of the Fermi National Accelerator Laboratory, where a large part of this work was completed, and thanks Josh Frieman and Brian Nord for kindly hosting. This paper is based upon work supported by the Visiting Scholars Award Program of the Universities Research Association. LLS was supported by the Science and Technology Facilities Council. HVP was partially supported by the European Research Council (ERC) under the European Community's Seventh Framework Programme (FP7/2007-2013)/ERC grant agreement number 306478- CosmicDawn, and the research project grant "Fundamental Physics from Cosmological Surveys" funded by the Swedish Research Council (VR) under Dnr 2017-04212. AP was supported by the Royal Society. This work was partially enabled by funding from the UCL Cosmoparticle Initiative.

6 Future work

6.1 Abstract

Future work will present an application to cosmological structure formation of the new field of research of *knowledge extraction* in machine learning. The aim is to extract physical knowledge about dark matter halo formation from the parameters learnt by a deep learning model, trained to infer final halo masses starting from the initial density field. To do this, we plan to develop a network architecture that combines convolutional neural networks with variational auto-encoders. The variational auto-encoder learns to compress the information in the initial conditions relevant to halo formation into a lower-dimensional representation, which can be interpreted in relation to physical aspects of the initial conditions. First, we will use this to investigate whether the features learnt by the deep learning model correlate with physically-meaningful features such as spherically-averaged overdensities. We then plan to extend the framework to gain new insights into the physics of halo formation from the learning of the model, thus improving our understanding of the relationship between the linear universe and the non-linear large-scale structure.

6.2 Knowledge extraction from the deep learning model

Machine learning methods, especially deep neural networks, are used widely both in industry and in scientific applications. Usually these models are trained to yield the best possible predictions, but recently there has also been increasing interest for understanding the way a specific model operates and the underlying reasons for the produced results. This is known as *interpretability* in machine learning (see Chapter 2 for a review on tools for interpretability in deep learning). Our main goal for utilizing machine learning is scientific understanding; we wish to gain new insights from a deep

learning model by extracting information from its learnt parameters and its outputs regarding the underlying physics of the problem of interest. This new field of research, called *knowledge extraction* from deep learning, has not yet been applied to cosmology.

Recently, there have been several applications in the field of natural sciences that use machine learning for scientific discovery (Roscher et al. 2019). For example, Iten et al. (2018) introduced `SciNet`, a modified variational auto-encoder (VAE; see Sec. 6.2.1 for an introduction to VAEs) which learns a representation from experimental data, that can be used to derive physical concepts. They applied their model to simple physical problems and demonstrated that the network is capable of finding physically-relevant parameters, as the physical parameters and those learnt by the machine learning model have a linear relationship. Similarly, Ye et al. (2018) constructed a low-dimensional representation encoding the physical parameters to predict the outcome of a collision of objects from videos. However, their works rely on prior knowledge about relevant parameters. For example, the architecture of the model adopted by Iten et al. (2018) was designed with prior knowledge about the underlying physical process. Moreover, the interpretability of their results was only possible thanks to the ability to compare to already known representations in physics.

Our ultimate aim is to gain physical insights in scenarios where the relevant physical parameters are not known a priori, but can be instead extracted from an interpretable deep learning model. Future work will focus on extracting new physical knowledge of non-linear halo collapse. This means using the deep learning model to extract the physical aspects of the initial density field that are most relevant to describe the formation of the final haloes. As a first goal, we plan to test whether the deep learning algorithm is, at a minimum, capable of finding physical features which we know are important for dark matter halo formation.

In Chapter 5, we used a convolutional neural network (CNN) to map the dark matter particles in the initial conditions onto the mass of the final haloes to which they belong at $z = 0$. We found that the predictions of the CNN match those of machine learning algorithms trained on information about spherical overdensities alone. This suggested that the CNN could be learning features that resemble those of spherical overdensities. On the other hand, this is only a hypothesis; the fact that the CNN and the GBTs return consistent predictions does not directly imply that the CNN is learning the same features used by the GBTs. It is possible that the CNN learns features that are uncorrelated with spherical overdensities, while still returning predictions that are consistent with those of the GBTs. Our next goal is to test this hypothesis. To do this, we require a deep learning model that is interpretable i.e., one which extracts relevant features that can be interpreted and related to physical aspects of the initial conditions, such as spherical overdensities. We plan to modify the

CNN architecture presented in Chapter 5 to one which combines CNNs with VAEs, similar to the architecture used in Iten et al. (2018). The VAE framework will allow us (i) to understand what features the algorithm is learning and (ii) to test our hypothesis about the correlation between the features learnt by the CNN and spherical overdensities. We give a general introduction to auto-encoders, and in particular VAEs, in Sec. 6.2.1 and then describe the deep learning architecture we plan to use for model interpretability in Sec. 6.2.2.

6.2.1 Variational auto-encoders

An auto-encoder is an unsupervised type of network that learns to compress the input data into a lower-dimensional representation, known as the latent representation, and then decompresses that to reconstruct something that is closely similar to the input data (Bourlard and Kamp 1988; Hinton and Zemel 1993). The former part of the algorithm is known as the *encoder* and the second as the *decoder*. The latent representation encodes the relevant information contained in the input data into the lowest possible dimensions. The network architecture for auto-encoders varies between simple feed-forward neural networks or convolutional neural networks depending on the use case; in the latter case the network is known as a convolutional auto-encoder (CAE). Typically, training an auto-encoder involves minimizing (via backpropagation) a reconstruction loss, measuring how well the decoder can reconstruct an output that is identical to the original input, starting from the latent representation. When the latent representation allows a good reconstruction of its input, then it has retained the most important information present in the input data.

Auto-encoders are generally effective at dimensionality reduction, feature learning, denoising images or generative modelling. However, one fundamental limitation of these algorithms is that the latent space they convert their inputs to may not be continuous, or allow easy interpolation. If the latent space has discontinuities, the decoder will return unrealistic outputs when sampling from any region that overlaps a discontinuity in latent space. This is because during training, the decoder was never given examples of encoded vectors coming from that region of latent space. VAEs are one family of auto-encoders that, by construction, yield continuous latent representations (Kingma and Welling 2014; Rezende et al. 2014). Instead of returning an encoded latent vector of size n, the encoder outputs two vectors of size n: a vector of means $\boldsymbol{\mu}$ and a vector of standard deviations $\boldsymbol{\sigma}$. The latent vector $\{z_i\}_1^n$, used as input to the decoder, is then given by random samples from Gaussian distributions of means $\boldsymbol{\mu}$ and standard deviations $\boldsymbol{\sigma}$ i.e., $z_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$. In this way, the auto-encoder learns the probability distribution functions over latent space, $p(\mathbf{z}|\mathbf{x})$, where $\mathbf{z}$ and

$\mathbf{x}$ are the latent and input variables respectively, resulting in a smooth latent space which can be easily interpolated. The loss function that is minimized when training a VAE is given by

$$\mathcal{L}_{\rm VAE} = ||\mathbf{x} - \hat{\mathbf{x}}||^2 - D_{\rm KL}(\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2) \parallel \mathcal{N}(\mathbf{0}, \mathbf{1})), \tag{6.1}$$

where the first term is a reconstruction term, that measures the performance of the encoding-decoding scheme, and the second is a regularization term, that regularizes the latent space by making the distributions returned by the encoder close to a standard normal distribution. Without the regularization term, the encoder can generate very different means $\boldsymbol{\mu}$ and small standard deviations $\boldsymbol{\sigma}$ as there are no limits to what values μ and σ can take. This would yield latent variables which are tightly clustered and far apart from each other, making interpolation difficult. Instead, the latent variables should be as close as possible to each other (while still being distinct), allowing smooth interpolation. The regularization term is introduced in the loss function to ensure that the latent space satisfies continuity i.e., two points close in latent space should return similar content once decoded, and completeness i.e., all points sampled in latent space should return meaningful content once decoded.

6.2.2 Convolutional neural networks with variational auto-encoders

As mentioned at the start of Sec. 6.2, our first goal will be to understand whether the features learnt by the CNN model used in Chapter 5 correlate with spherical overdensities. To do this, we plan to use VAEs to compress the initial density field into a lower-dimensional set of latent variables, which contain relevant information to infer the mass of the final haloes at $z = 0$. The latent variables can then be directly compared to physically-meaningful features, such as spherical overdensities, to test whether or not there is a correlation between the two.

Traditional VAEs are *unsupervised* neural networks that only learn to encode and decode the input data and therefore would have no information about the final haloes when learning to compress the initial conditions field into latent variables. In order to obtain a set of latent variables that do contain information about halo mass, we propose a deep learning architecture that combines variational auto-encoders for data compression with fully-connected layers for the halo mass predictions, as illustrated in Fig. 6.1. First, we will start from the CNN model used in Chapter 5. The model, trained on N-body simulations, takes the 3D initial density field centred at the initial position of a dark matter particle, and returns a prediction for the mass of the halo to which that particle belongs at $z = 0$ (top panel of Fig. 6.1). Since we are interested in interpreting the features that lead to

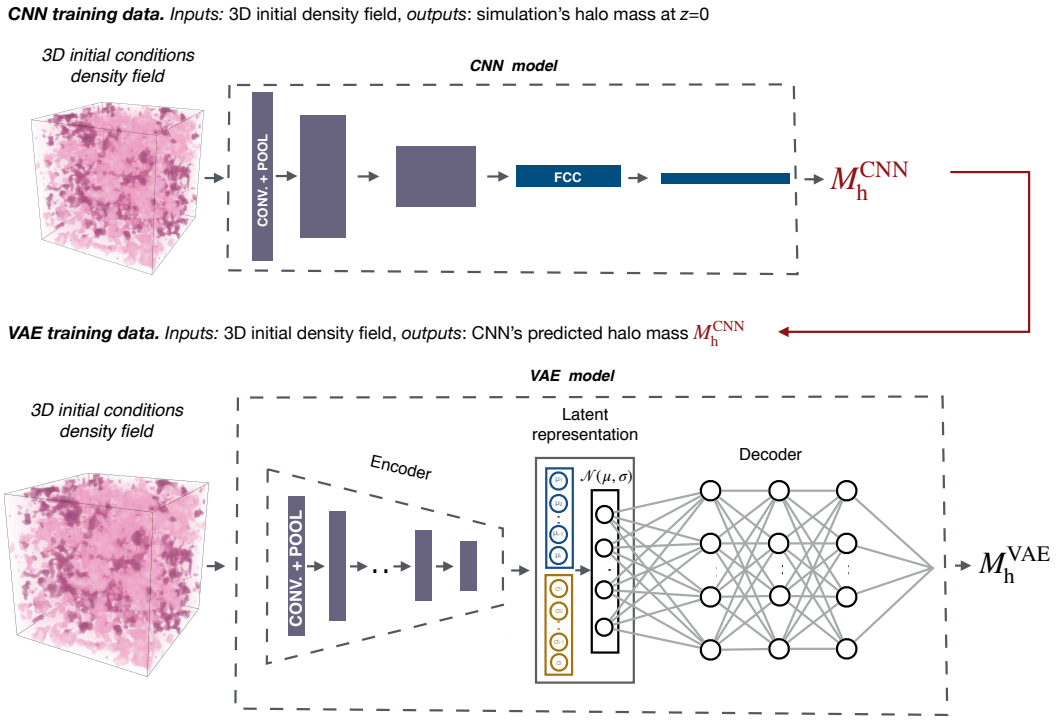

Figure 6.1: We plan to develop a VAE-like model (*bottom panel*) that will allow us to interpret the features learnt by the CNN adopted in Chapter 5 (*top panel*) in relation to known physical aspects of the initial conditions, such as spherical overdensities. The CNN is trained on N-body simulations to predict the final mass of the halo to which a dark matter particle belongs at $z = 0$, starting from the 3D initial density field. The CNN predictions are used as ground truth labels to a VAE model, made of an encoder and a decoder that outputs halo mass. The encoder compresses the information in the initial conditions relevant to the final halo mass into two vectors, one of means μ and another of standard deviations σ. The latent vector is then randomly-sampled from Gaussian distributions of those means and standard deviations. We will then be able to answer the question of whether the CNN extracts features that resemble spherical overdensities by measuring the correlation between the latent variables and spherical overdensities.

the predictions made by the CNN, we will use the CNN-predicted halo mass as the ground truth label for the VAE model. In other words, the training data of the VAE model consist of the 3D initial conditions density field as input and the halo mass predicted by the CNN model for each dark matter particle as output. This setup will allow us to directly recover the features that lead to the predictions returned by the CNN.

The architecture of the proposed VAE model is shown in the bottom panel of Fig. 6.1. The first part is that of the encoder, where the algorithm learns the compression. The encoder consists of convolutional and pooling layers similar to the CNN architecture used in Chapter 5, without fully-connected layers. The latent variables are then used as inputs to the fully-connected layers, which output the (CNN predicted) mass of the halo associated with each particle. Since the ground truth label of the VAE model is given by the halo mass predictions of the CNN, the loss function minimized by the VAE network is given by

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\left(M_{i,\mathrm{h}}^{\mathrm{CNN}} - M_{i,\mathrm{h}}^{\mathrm{VAE}}\right)^2 - D_{\mathrm{KL}}(\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\sigma}^2)\parallel\mathcal{N}(\mathbf{0},\mathbf{1})), \tag{6.2}$$

where N is the number of training samples and $M_{i,\mathrm{h}}^{\mathrm{CNN}}$ and $M_{i,\mathrm{h}}^{\mathrm{VAE}}$ are the predicted masses of the CNN and of the VAE for particle i, respectively. The first term in Eq. 6.2 is the mean squared error, which measures how well the model predicts halo mass, and the second term is the regularization term, which forces the latent space distributions to be close to normal distributions (see Sec. 6.2.1). The weights of the architecture are learnt by minimizing the loss in Eq. (6.2) via standard backpropagation. In this way, the encoder learns to compress the initial density field into a set of latent variables which contain the information required to predict final halo masses. Since the VAE and the CNN models are intended to be similar models, with the exception of the additional latent layer in the VAE, we fix the values of the hyperparameters of the VAE to those used for the CNN.

One important additional hyperparameter in the VAE model that must be set prior to training is the dimensionality of the latent space. In Chapter 4, we found that the gradient-boosted trees learn predominantly learns from 7 features: the values of the density field smoothed with a top-hat window function on seven mass scales evenly spaced in $\log M$ within the range $13 \leq \log(M_{\mathrm{smoothing}}/\mathrm{M}_\odot) \leq 14$. Therefore, we will start with a choice of seven latent variables and test whether this choice is sufficient to capture the information required to yield halo mass predictions that match those of the CNN. We expect further investigation on the impact of our choice of dimensionality of the latent space on the performance of the VAE model. The final step will be to quantify the correlation between the latent variables and the seven spherical overdensity features using a covariance matrix. This will ultimately allow us to verify whether the algorithm is learning physically-meaningful information and whether this information correlates with overdensities smoothed on mass scales $13 \leq \log(M_{\mathrm{smoothing}}/\mathrm{M}_\odot) \leq 14$. This is similar to what was done in Iten et al. (2018), where they concluded that their model learnt the physically-relevant parameters given the linear relationship

between the physical parameters and the latent variables.

Overall, the deep learning framework illustrated in Fig. 6.1 learns about halo formation starting from the initial density field, while simultaneously returning a set of latent variables which compresses the information in the initial conditions that is useful for making halo mass predictions. This framework is not limited to investigating correlations between known physical parameters and latent variables. We plan to further expand this framework to extract new physical relations between the initial conditions and final dark matter haloes, which may include localised components around peaks in the initial density field or large-scale flows.

7 Conclusions

In this thesis, we presented the first applications of a new machine learning approach to gain physical understanding of cosmological structure formation. The approach consists of training a machine learning algorithm to learn the non-linear relationship between the initial conditions and the final mass of dark matter haloes directly from N-body simulations. The strength of our method lies in its ability to establish a physical interpretation of the machine learning results. In this way, we were able to learn about which physical aspects of the early universe contain relevant information about the final dark matter haloes.

We investigated the impact of different properties of the initial linear fields on the formation of dark matter haloes. We started with a machine learning binary classification framework (Chapter 3), where a random forest model was trained to predict whether dark matter particles will collapse into haloes of mass above or below a threshold $M_{\rm th} = 1.8 \times 10^{12}\,{\rm M}_{\odot}$. Contrary to existing interpretations of the Sheth-Tormen ellipsoidal collapse model, we found that the tidal shear field does not contain additional information over that contained in the density field for our classification problem. This result was confirmed by quantitatively showing that the learning process of the machine learning algorithm is predominantly driven by the local overdensity around dark matter particles on smoothing scales $10^{12} \leq M_{\rm smoothing}/{\rm M}_{\odot} \leq 10^{14}$ and is unaffected by the surrounding tidal shear. By comparing the machine learning predictions with those of analytic theories, we found that the linear density field contains sufficient information to yield predictions at the accuracy level of both spherical and ellipsoidal collapse analytic frameworks. Therefore, the machine learning and analytic frameworks consistently show that the tidal shear field carries little role in the formation of dark matter haloes around this mass threshold.

This result generalized to a regression setting (Chapter 4), where a gradient boosted trees model was trained to infer the final mass of the halo to which a dark matter particle will belong

at $z = 0$. We found no significant impact of the tidal shear field on the final mass of haloes in the range $11.4 \leq \log(M/\mathrm{M}_{\odot}) \leq 13.4$. We quantified this with a machine learning model comparison using a metric based on the Kullback-Leibler divergence. The addition of tidal shear information does not yield an improved halo collapse model over one based on density information alone, as the difference in the predictive performance of the two models is consistent with the statistical uncertainty of the density-only based model.

The work presented in Chapters 3 $\&$ 4 was based on machine learning models that require their inputs to be hand-crafted features. This approach is therefore limited by our ability to put forward meaningful features, which are motivated by incomplete analytic approximations and can only capture limited aspects of the initial conditions. To go beyond this, we extended our framework to convolutional neural networks (CNNs; Chapter 5), as these can learn about halo formation directly from the linear density field realization, without the need for feature extraction. We compared the performance of the CNN to that of the tree ensemble models adopted in Chapters 3 $\&$ 4. Despite the fact that the inputs to the CNN contain all the information necessary to fully describe the initial conditions of the Universe, the CNN's predictions are consistent with those of the ensembles of trees, whose learning is based only on spherically-averaged overdensities. Surprisingly, this may indicate that spherical overdensities saturate the most relevant information to halo formation contained in the initial conditions. In Chapter 6, we outlined the next steps towards the ultimate goal of this work: developing an interpretable deep learning framework that can be used to extract new physical knowledge of the relation between the initial conditions and the final dark matter haloes. At first, this will involve interpreting the information learnt by the CNN model with respect to any relevant physical aspects of the initial conditions, such as spherical overdensities. We plan to do this by adopting a hybrid architecture that combines convolutional neural networks with variational auto-encoders.

The work presented in this thesis points toward a new field of research, knowledge extraction from machine learning, which utilizes interpretable machine learning frameworks for the purpose of extracting new physical knowledge about cosmological structure formation.

A

Appendix to Chapter 4

A.1 A comparison with analytic theories

We validated the machine learning findings by comparing the accuracy of its predictions against those of analytic theories which also provide final halo mass predictions based on the same initial conditions information.

We compared the machine learning predictions based on the density features with EPS theory and those based on density and tidal shear features with ST theory, for the test set particles in *sim*-1. According to EPS, the fraction of density trajectories with a first upcrossing of a density threshold barrier $\delta_{\rm th}$ is equivalent to the fraction of haloes of mass M. The density threshold barrier $\delta_{\rm th}$ adopted by Bond et al. (1991) is that of spherical collapse: $\delta_{\rm th}(z) = (D(z)/D(0))\delta_{\rm sc}$, where $\delta_{\rm sc} \approx 1.686$. The predicted halo mass of each test particle is given by the smoothing mass scale at which the particle first upcrosses the density threshold barrier.

In the ST formalism, EPS theory is extended to account for the effect of the tidal shear field by adopting a "moving" collapse barrier rather than the spherical collapse barrier. The ST collapse barrier $b(z)$ varies as a function of the mass variance $\sigma^2(M)$ and is given by

$$b(z) = \sqrt{a}\delta_{\rm sc}(z)\left[1 + \left(\beta\frac{\sigma^2(M)}{a\delta_{\rm sc}^2(z)}\right)^{\gamma}\right], \tag{A.1}$$

where $\delta_{\rm sc}(0) \approx 1.686$ and the best-fit parameters found in Sheth et al. (2001) are $\beta = 0.485$, $\gamma = 0.615$ and $a = 0.707$. Similar to the EPS case, the predicted halo mass of each test particle is given by the smoothing mass scale at which the particle first upcrosses the threshold barrier given by Eq. (A.1). In summary, for each test particles we can compute the EPS and ST predicted halo masses and compare those to the machine learning density-only and density combined with shear

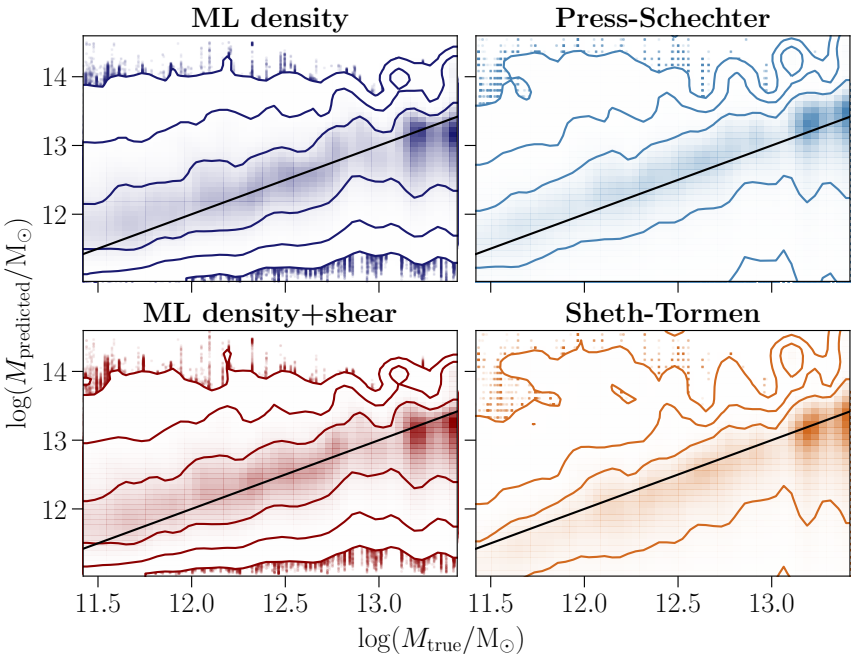

Figure A.1: Two-dimensional histograms and contours containing 68%, 95% and 99.7% of the joint probability of the predicted vs. true halo masses for the analytic and machine learning models. We compare the machine learning predictions based on the density features with EPS theory and those based on density and tidal shear features with ST theory. The predictions are qualitatively similar, but with tighter confidence regions in the machine learning case. This validates our machine learning results as we find no evidence of any relevant information contained in the features that the algorithm fails to learn.

predictions, respectively.

Figure A.1 shows the predicted halo masses as a function of true halo masses for the analytic and machine learning models. We show two-dimensional histograms and the contours containing 68%, 95% and 99.7% of the joint probability. Machine learning and analytic models show qualitatively similar predictions, but with tighter confidence regions for the machine learning predictions. This is especially notable where the analytic models' predictions extend to much lower mass values than the machine learning predictions. Note also that the ST predictions are shifted towards lower mass values compared to the PS predictions, for fixed true halo mass. This directly reflects the fact that the ST collapse barrier takes larger δ values than the PS barrier at fixed smoothing mass scale; the same particle will therefore cross the collapse barrier at lower smoothing mass scales for ST than PS, which in turn results in a lower halo mass prediction.

This test validates our machine learning results by ruling out the possibility that the algorithm

is not making use of all the information contained in the features. Moreover, this also shows that the machine learning algorithm provides better predictions than the analytic models on a particle-by-particle basis.

In Sec. 4.5, we compared the performance of two machine learning models by computing the KL divergence between the distributions of number of particles within bins of halo mass predicted by the two models. A similar quantitative comparison between the machine learning and the analytic models is not possible. This is because in order to compute the KL divergence between two distributions, these must share the same normalization. Instead, the total number of particles in halos predicted by the machine learning algorithm is not the same as that predicted by the analytic theories. The former is given by all the particles in the test set, whereas the latter is given by only those particles in the test set that cross the collapse thresholds. Therefore, one cannot use the KL divergence to meaningfully quantify the difference between the machine learning and the analytic predicted distributions. A qualitative comparison between the distributions as in Fig. A.1 is sufficient for the purpose of this work.

Bibliography

Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin. Learning from data. 2012.

S. Agarwal, R. Dav, and B. A. Bassett. Painting galaxies into dark matter haloes using machine learning. *Monthly Notices of the Royal Astronomical Society*, 478(3):34103422, May 2018. ISSN 1365-2966. doi: 10.1093/mnras/sty1169. URL http://dx.doi.org/10.1093/mnras/sty1169.

A. Albrecht and P. J. Steinhardt. Cosmology for grand unified theories with radiatively induced symmetry breaking. *Phys. Rev. Lett.*, 48:1220–1223, Apr 1982. doi: 10.1103/PhysRevLett.48.1220. URL https://link.aps.org/doi/10.1103/PhysRevLett.48.1220.

R. A. Alpher, H. Bethe, and G. Gamow. The Origin of Chemical Elements. *Physical Review*, 73: 803–804, Apr. 1948. doi: 10.1103/PhysRev.73.803.

L. Anderson, E. Aubourg, S. Bailey, F. Beutler, A. S. Bolton, J. Brinkmann, J. R. Brownstein, C.-H. Chuang, A. J. Cuesta, K. S. Dawson, D. J. Eisenstein, S. Ho, K. Honscheid, E. A. Kazin, D. Kirkby, M. Manera, C. K. McBride, O. Mena, R. C. Nichol, M. D. Olmstead, N. Padmanabhan, N. Palanque-Delabrouille, W. J. Percival, F. Prada, A. J. Ross, N. P. Ross, A. G. Sánchez, L. Samushia, D. J. Schlegel, D. P. Schneider, H.-J. Seo, M. A. Strauss, D. Thomas, J. L. Tinker, R. Tojeiro, L. Verde, D. Wake, D. H. Weinberg, X. Xu, and C. Yeche. The clustering of galaxies in the SDSS-III Baryon Oscillation Spectroscopic Survey: measuring D_A and H at z = 0.57 from the baryon acoustic peak in the Data Release 9 spectroscopic Galaxy sample. MNRAS, 439:83–101, Mar. 2014. doi: 10.1093/mnras/stt2206.

M. A. Aragon-Calvo. Classifying the large-scale structure of the universe with deep neural networks. MNRAS, 484(4):5771–5784, Apr 2019. doi: 10.1093/mnras/stz393.

S. Avila, A. Knebe, F. R. Pearce, A. Schneider, C. Srisawat, P. A. Thomas, P. Behroozi, P. J. Elahi, J. Han, Y.-Y. Mao, J. Onions, V. Rodriguez-Gomez, and D. Tweed. SUSSING MERGER TREES: the influence of the halo finder. MNRAS, 441(4):3488–3501, Jul 2014. doi: 10.1093/mnras/stu799.

N. M. Ball and R. J. Brunner. Data Mining and Machine Learning in Astronomy. *International Journal of Modern Physics D*, 19(7):1049–1106, Jan 2010. doi: 10.1142/S0218271810017160.

N. M. Ball, J. Loveday, M. Fukugita, O. Nakamura, S. Okamura, J. Brinkmann, and R. J. Brunner. Galaxy types in the Sloan Digital Sky Survey using supervised artificial neural networks. *Mon. Not. Roy. Astron. Soc.*, 348:1038, 2004. doi: 10.1111/j.1365-2966.2004.07429.x.

N. M. Ball, R. J. Brunner, A. D. Myers, N. E. Strand, S. L. Alberts, and D. Tcheng. Robust machine learning applied to astronomical data sets. iii. probabilistic photometric redshifts for galaxies and quasars in the sdss and galex. *The Astrophysical Journal*, 683(1):12, 2008.

M. Banerji, O. Lahav, C. J. Lintott, F. B. Abdalla, K. Schawinski, S. P. Bamford, D. Andreescu, P. Murray, M. J. Raddick, A. Slosar, and et al. Galaxy zoo: reproducing galaxy morphologies via machine learning. *Monthly Notices of the Royal Astronomical Society*, 406(1):342353, Apr 2010. ISSN 0035-8711. doi: 10.1111/j.1365-2966.2010.16713.x. URL http://dx.doi.org/10.1111/j.1365-2966.2010.16713.x.

J. M. Bardeen, J. R. Bond, N. Kaiser, and A. S. Szalay. The statistics of peaks of Gaussian random fields. ApJ, 304:15–61, May 1986. doi: 10.1086/164143.

J. Barnes and P. Hut. A hierarchical O(N log N) force-calculation algorithm. Nature, 324:446–449, Dec. 1986. doi: 10.1038/324446a0.

D. Baumann. Inflation. In *Physics of the large and the small, TASI 09, proceedings of the Theoretical Advanced Study Institute in Elementary Particle Physics, Boulder, Colorado, USA, 1-26 June 2009*, pages 523–686, 2011. doi: 10.1142/9789814327183_0010.

Y. Bengio. Learning deep architectures for ai. *Foundations and Trends in Machine Learning*, 2(1): 1–127, 2009. ISSN 1935-8237. doi: 10.1561/2200000006. URL http://dx.doi.org/10.1561/2200000006.

C. L. Bennett, M. Halpern, G. Hinshaw, N. Jarosik, A. Kogut, M. Limon, S. S. Meyer, L. Page, D. N. Spergel, G. S. Tucker, and et al. Firstyearwilkinson microwave anisotropy probe(wmap) observations: Preliminary maps and basic results. *The Astrophysical Journal Supplement Series*, 148(1):127, Sep 2003. ISSN 1538-4365. doi: 10.1086/377253. URL http://dx.doi.org/10.1086/377253.

P. Berger and G. Stein. A volumetric deep Convolutional Neural Network for simulation of mock dark matter halo catalogues. MNRAS, 482(3):2861–2871, Jan 2019. doi: 10.1093/mnras/sty2949.

J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13(1):281–305, Feb. 2012. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=2503308.2188395.

E. Bertin. Classification of astronomical images with a neural network. In N. Epchtein, A. Omont, B. Burton, and P. Persi, editors, *Science with Astronomical Near-Infrared Sky Surveys*, pages 49–51, Dordrecht, 1994. Springer Netherlands. ISBN 978-94-011-0946-8.

E. Bertin and S. Arnouts. SExtractor: Software for source extraction. A&AS, 117:393–404, June 1996. doi: 10.1051/aas:1996164.

G. Bertone and D. Hooper. History of dark matter. *Rev. Mod. Phys.*, 90:045002, Oct 2018. doi: 10.1103/RevModPhys.90.045002. URL https://link.aps.org/doi/10.1103/RevModPhys.90.045002.

C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., New York, NY, USA, 1995. ISBN 0198538642.

C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387310738.

J. R. Bond and S. T. Myers. The Peak-Patch Picture of Cosmic Catalogs. I. Algorithms. ApJS, 103:1, Mar. 1996. doi: 10.1086/192267.

J. R. Bond, S. Cole, G. Efstathiou, and N. Kaiser. Excursion set mass functions for hierarchical Gaussian fluctuations. ApJ, 379:440–460, Oct. 1991. doi: 10.1086/170520.

M. Borzyszkowski, A. D. Ludlow, and C. Porciani. The formation of cold dark matter haloes - II. Collapse time and tides. MNRAS, 445:4124–4136, Dec. 2014. doi: 10.1093/mnras/stu2033.

H. Bourlard and Y. Kamp. Auto-association by multilayer perceptrons and singular value decomposition. *Biological Cybernetics*, 59:291–294, 1988.

L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. ISSN 1573-0565. doi: 10.1023/A:1010933404324.

L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. Classification and regression trees, 1984.

S. Carliles, T. Budavari, S. Heinis, C. Priebe, and A. Szalay. Photometric Redshift Estimation on SDSS Data Using Random Forests. *ASP Conf. Ser.*, 394:521, 2008.

B. Carr, M. Raidal, T. Tenkanen, V. Vaskonen, and H. Veermäe. Primordial black hole constraints for extended mass functions. Phys. Rev. D, 96(2):023514, Jul 2017. doi: 10.1103/PhysRevD.96.023514.

B. J. Carr and S. W. Hawking. Black Holes in the Early Universe. *Monthly Notices of the Royal Astronomical Society*, 168(2):399–415, 08 1974. ISSN 0035-8711. doi: 10.1093/mnras/168.2.399. URL `https://doi.org/10.1093/mnras/168.2.399`.

R. Caruana and A. Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 161–168, New York, NY, USA, 2006. ACM. ISBN 1-59593-383-2. doi: 10.1145/1143844.1143865.

T. Charnock, G. Lavaux, B. D. Wandelt, S. Sarma Boruah, J. Jasche, and M. J. Hudson. Neural physical engines for inferring the halo mass distribution function. *arXiv e-prints*, art. arXiv:1909.06379, Sep 2019.

N. E. Chisari and M. Zaldarriaga. Connection between Newtonian simulations and general relativity. Phys. Rev. D, 83(12):123505, Jun 2011. doi: 10.1103/PhysRevD.83.123505.

K. Clark, M.-T. Luong, C. D. Manning, and Q. Le. Semi-supervised sequence modeling with cross-view training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1914–1925, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1217. URL `https://www.aclweb.org/anthology/D18-1217`.

D. Clowe, M. Bradač, A. H. Gonzalez, M. Markevitch, S. W. Randall, C. Jones, and D. Zaritsky. A Direct Empirical Proof of the Existence of Dark Matter. ApJ, 648(2):L109–L113, Sep 2006. doi: 10.1086/508162.

A. A. Collister and O. Lahav. ANNz: Estimating Photometric Redshifts Using Artificial Neural Networks. PASP, 116:345–351, Apr. 2004. doi: 10.1086/383254.

S. Das, T. Louis, M. R. Nolta, G. E. Addison, E. S. Battistelli, J. R. Bond, E. Calabrese, D. Crichton, M. J. Devlin, S. Dicker, J. Dunkley, R. Dünner, J. W. Fowler, M. Gralla, A. Hajian, M. Halpern, M. Hasselfield, M. Hilton, A. D. Hincks, R. Hlozek, K. M. Huffenberger, J. P. Hughes, K. D. Irwin, A. Kosowsky, R. H. Lupton, T. A. Marriage, D. Marsden, F. Menanteau, K. Moodley, M. D. Niemack, L. A. Page, B. Partridge, E. D. Reese, B. L. Schmitt, N. Sehgal, B. D. Sherwin, J. L. Sievers, D. N. Spergel, S. T. Staggs, D. S. Swetz, E. R. Switzer, R. Thornton, H. Trac, and E. Wollack. The Atacama Cosmology Telescope: temperature and gravitational lensing power spectrum measurements from three seasons of data. J. Cosmology Astropart. Phys., 2014(4):014, Apr 2014. doi: 10.1088/1475-7516/2014/04/014.

M. Davis, J. Huchra, D. W. Latham, and J. Tonry. A survey of galaxy redshifts. II - The large scale space distribution. ApJ, 253:423–445, Feb. 1982. doi: 10.1086/159646.

W. Dehnen. Towards optimal softening in three-dimensional N-body codes - I. Minimizing the force error. MNRAS, 324(2):273–291, Jun 2001. doi: 10.1046/j.1365-8711.2001.04237.x.

L. Deng and D. Yu. Deep learning: Methods and applications. *Foundations and Trends in Signal Processing*, 7(34):197–387, 2014. ISSN 1932-8346. doi: 10.1561/2000000039. URL http://dx.doi.org/10.1561/2000000039.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv e-prints*, art. arXiv:1810.04805, Oct 2018.

A. Diba, V. Sharma, A. Mohammad Pazandeh, H. Pirsiavash, and L. Van Gool. Weakly supervised cascaded convolutional networks. 07 2017. doi: 10.1109/CVPR.2017.545.

R. H. Dicke, P. J. E. Peebles, P. G. Roll, and D. T. Wilkinson. Cosmic Black-Body Radiation. ApJ, 142:414–419, July 1965. doi: 10.1086/148306.

T. G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Mach. Learn.*, 40(2):139–157, Aug. 2000a. ISSN 0885-6125. doi: 10.1023/A:1007607513941. URL https://doi.org/10.1023/A:1007607513941.

T. G. Dietterich. Ensemble methods in machine learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems*, MCS '00, pages 1–15, London, UK, UK, 2000b. Springer-Verlag. ISBN 3-540-67704-6. URL http://dl.acm.org/citation.cfm?id=648054.743935.

S. Dodelson. *Modern Cosmology*. Academic Press, Burlington, 2003.

S. Dodelson and L. M. Widrow. Sterile neutrinos as dark matter. Phys. Rev. Lett., 72(1):17–20, Jan 1994. doi: 10.1103/PhysRevLett.72.17.

A. G. Doroshkevich. The space structure of perturbations and the origin of rotation of galaxies in the theory of fluctuation. *Astrofizika*, 6:581–600, 1970.

P. Douglas, S. Harris, A. Yuille, and M. S. Cohen. Performance comparison of machine learning algorithms and number of independent components used in fmri decoding of belief vs. disbelief. *NeuroImage*, 56(2):544 – 553, 2011. ISSN 1053-8119. doi: https://doi.org/10.1016/j.neuroimage.2010.11.002. URL http://www.sciencedirect.com/science/article/pii/S105381191001414X. Multivariate Decoding and Brain Reading.

C. Dreissigacker, R. Sharma, C. Messenger, R. Zhao, and R. Prix. Deep-learning continuous gravitational waves. Phys. Rev. D, 100(4):044009, Aug 2019. doi: 10.1103/PhysRevD.100.044009.

V. Dumoulin and F. Visin. A guide to convolution arithmetic for deep learning. *arXiv e-prints*, art. arXiv:1603.07285, Mar 2016.

J. Dunkley, E. Komatsu, M. R. Nolta, D. N. Spergel, D. Larson, G. Hinshaw, L. Page, C. L. Bennett, B. Gold, N. Jarosik, J. L. Weiland, M. Halpern, R. S. Hill, A. Kogut, M. Limon, S. S. Meyer, G. S. Tucker, E. Wollack, and E. L. Wright. Five-Year Wilkinson Microwave Anisotropy Probe Observations: Likelihoods and Parameters from the WMAP Data. ApJS, 180:306–329, Feb. 2009. doi: 10.1088/0067-0049/180/2/306.

A. Einstein. Kosmologische Betrachtungen zur allgemeinen Relativitätstheorie. *Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften (Berlin*, pages 142–152, Jan 1917.

O. Fakhouri, C.-P. Ma, and M. Boylan-Kolchin. The merger rates and mass assembly histories of dark matter haloes in the two Millennium simulations. MNRAS, 406(4):2267–2278, Aug 2010. doi: 10.1111/j.1365-2966.2010.16859.x.

A. Farahi and A. J. Benson. Excursion set theory for correlated random walks. MNRAS, 433:3428–3439, Aug. 2013. doi: 10.1093/mnras/stt987.

T. Fawcett. An introduction to roc analysis. *Pattern Recogn. Lett.*, 27(8):861–874, June 2006. ISSN 0167-8655. doi: 10.1016/j.patrec.2005.10.010. URL http://dx.doi.org/10.1016/j.patrec.2005.10.010.

S. M. Feeney, H. V. Peiris, A. R. Williamson, S. M. Nissanke, D. J. Mortlock, J. Alsing, and D. Scolnic. Prospects for Resolving the Hubble Constant Tension with Standard Sirens. Phys. Rev. Lett., 122 (6):061105, Feb 2019. doi: 10.1103/PhysRevLett.122.061105.

A. E. Firth, O. Lahav, and R. S. Somerville. Estimating photometric redshifts with artificial neural networks. MNRAS, 339:1195–1202, Mar. 2003. doi: 10.1046/j.1365-8711.2003.06271.x.

S. R. Folkes, O. Lahav, and S. J. Maddox. An artificial neural network approach to the classification of galaxy spectra. MNRAS, 283:651–665, Dec. 1996. doi: 10.1093/mnras/283.2.651.

R. Fong and A. Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. pages 3449–3457, 10 2017. doi: 10.1109/ICCV.2017.371.

W. L. Freedman, B. F. Madore, D. Hatt, T. J. Hoyt, I. S. Jang, R. L. Beaton, C. R. Burns, M. G. Lee, A. J. Monson, J. R. Neeley, M. M. Phillips, J. A. Rich, and M. Seibert. The Carnegie-Chicago Hubble Program. VIII. An Independent Determination of the Hubble Constant Based on the Tip of the Red Giant Branch. ApJ, 882(1):34, Sep 2019. doi: 10.3847/1538-4357/ab2f73.

K. C. Freeman. On the Disks of Spiral and S0 Galaxies. ApJ, 160:811, June 1970. doi: 10.1086/150474.

Y. Freund and R. E. Schapire. Game theory, on-line prediction and boosting. In *Proceedings of the Ninth Annual Conference on Computational Learning Theory*, COLT '96, pages 325–332, New York, NY, USA, 1996. ACM. ISBN 0-89791-811-8. doi: 10.1145/238061.238163. URL http://doi.acm.org/10.1145/238061.238163.

Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

J. H. Friedman. Greedy function approximation: A gradient boosting machine. *Ann. Statist.*, 29(5): 1189–1232, 10 2001. doi: 10.1214/aos/1013203451. URL https://doi.org/10.1214/aos/1013203451.

J. H. Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4): 367 – 378, 2002. ISSN 0167-9473. doi: https://doi.org/10.1016/S0167-9473(01)00065-2. URL http://www.sciencedirect.com/science/article/pii/S0167947301000652. Nonlinear Methods and Data Mining.

T. D. Gebhard, N. Kilbertus, I. Harry, and B. Schölkopf. Convolutional neural networks: A magic bullet for gravitational-wave detection? Phys. Rev. D, 100(6):063015, Sep 2019. doi: 10.1103/PhysRevD.100.063015.

S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4:1–58, 01 1992. doi: 10.1162/neco.1992.4.1.1.

S. Genel, R. Genzel, N. Bouche, T. Naab, and A. Sternberg. The halo merger rate in the Millennium Simulation and implications for observed galaxy merger fractions. *Astrophys. J.*, 701:2002–2018, 2009. doi: 10.1088/0004-637X/701/2/2002.

E. M. George, C. L. Reichardt, K. A. Aird, B. A. Benson, L. E. Bleem, J. E. Carlstrom, C. L. Chang, H. M. Cho, T. M. Crawford, A. T. Crites, T. de Haan, M. A. Dobbs, J. Dudley, N. W. Halverson, N. L. Harrington, G. P. Holder, W. L. Holzapfel, Z. Hou, J. D. Hrubes, R. Keisler, L. Knox, A. T. Lee, E. M. Leitch, M. Lueker, D. Luong-Van, J. J. McMahon, J. Mehl, S. S. Meyer, M. Millea, L. M. Mocanu, J. J. Mohr, T. E. Montroy, S. Padin, T. Plagge, C. Pryke, J. E. Ruhl, K. K. Schaffer, L. Shaw, E. Shirokoff, H. G. Spieler, Z. Staniszewski, A. A. Stark, K. T. Story, A. van Engelen,

K. Vanderlinde, J. D. Vieira, R. Williamson, and O. Zahn. A Measurement of Secondary Cosmic Microwave Background Anisotropies from the 2500 Square-degree SPT-SZ Survey. ApJ, 799(2): 177, Feb 2015. doi: 10.1088/0004-637X/799/2/177.

S. P. D. Gill, A. Knebe, and B. K. Gibson. The evolution of substructure - I. A new identification method. MNRAS, 351:399–409, June 2004. doi: 10.1111/j.1365-2966.2004.07786.x.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. 2016.

D. Green and J. Swets. Signal detection theory and psychophysics, 1966.

S. R. Green and R. M. Wald. Newtonian and relativistic cosmologies. Phys. Rev. D, 85(6):063512, Mar 2012. doi: 10.1103/PhysRevD.85.063512.

M. Grossi, L. Verde, C. Carbone, K. Dolag, E. Branchini, F. Iannuzzi, S. Matarrese, and L. Moscardini. Large-scale non-Gaussian mass function and halo bias: tests on N-body simulations. MNRAS, 398:321–332, Sept. 2009. doi: 10.1111/j.1365-2966.2009.15150.x.

A. Gupta, J. M. Z. Matilla, D. Hsu, and Z. Haiman. Non-Gaussian information from weak lensing data via deep learning. Phys. Rev. D, 97(10):103515, May 2018. doi: 10.1103/PhysRevD.97.103515.

A. H. Guth. Inflationary universe: A possible solution to the horizon and flatness problems. *Phys. Rev. D*, 23:347–356, Jan 1981. doi: 10.1103/PhysRevD.23.347. URL https://link.aps.org/doi/10.1103/PhysRevD.23.347.

A. H. Guth and S. Y. Pi. Fluctuations in the New Inflationary Universe. *Phys. Rev. Lett.*, 49:1110–1113, 1982. doi: 10.1103/PhysRevLett.49.1110.

T. J. Hastie, R. Tibshirani, and J. H. Friedman. The elements of statistical learning: Data mining, inference, and prediction, 2nd edition. In *Springer Series in Statistics*, 2005.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2015.

S. He, Y. Li, Y. Feng, S. Ho, S. Ravanbakhsh, W. Chen, and B. Póczos. Learning to predict the cosmological structure formation. *Proceedings of the National Academy of Science*, 116(28): 13825–13832, Jul 2019. doi: 10.1073/pnas.1821458116.

J. Hilden. The area under the roc curve and its competitors. *Medical Decision Making*, 11 (2):95–101, 1991. doi: 10.1177/0272989X9101100204. URL https://doi.org/10.1177/0272989X9101100204. PMID: 1865785.

G. E. Hinton and R. S. Zemel. Autoencoders, minimum description length and helmholtz free energy. In *NIPS*, 1993.

G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural Comput.*, 18(7):1527–1554, July 2006. ISSN 0899-7667. doi: 10.1162/neco.2006.18.7.1527. URL http://dx.doi.org/10.1162/neco.2006.18.7.1527.

R. W. Hockney and J. W. Eastwood. *Computer simulation using particles*. 1988.

E. Hubble. A relation between distance and radial velocity among extra-galactic nebulae. *Proceedings of the National Academy of Sciences*, 15(3):168–173, 1929. ISSN 0027-8424. doi: 10.1073/pnas.15.3.168. URL https://www.pnas.org/content/15/3/168.

D. Huterer and D. L. Shafer. Dark energy two decades after: observables, probes, consistency tests. *Reports on Progress in Physics*, 81(1):016901, Jan 2018. doi: 10.1088/1361-6633/aa997e.

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 448–456. JMLR.org, 2015. URL http://dl.acm.org/citation.cfm?id=3045118.3045167.

R. Iten, T. Metger, H. Wilming, L. del Rio, and R. Renner. Discovering physical concepts with neural networks. *arXiv e-prints*, art. arXiv:1807.10300, Jul 2018.

S. J. Reddi, S. Kale, and S. Kumar. On the convergence of adam & beyond. 05 2018.

G. James, D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated, 2014. ISBN 1461471370, 9781461471370.

N. Jeffrey, F. Lanusse, O. Lahav, and J.-L. Starck. Deep learning dark matter map reconstructions from DES SV weak lensing data. *arXiv e-prints*, art. arXiv:1908.00543, Aug 2019.

A. Jenkins, C. S. Frenk, S. D. M. White, J. M. Colberg, S. Cole, A. E. Evrard, H. M. P. Couchman, and N. Yoshida. The mass function of dark matter haloes. MNRAS, 321:372–384, Feb. 2001. doi: 10.1046/j.1365-8711.2001.04029.x.

R. Jozefowicz, O. Vinyals, M. Schuster, N. Shazeer, and Y. Wu. Exploring the Limits of Language Modeling. *arXiv e-prints*, art. arXiv:1602.02410, Feb 2016.

H. M. Kamdar, M. J. Turk, and R. J. Brunner. Machine learning and cosmological simulations - I. Semi-analytical models. MNRAS, 455:642–658, Jan. 2016. doi: 10.1093/mnras/stv2310.

K. Kamnitsas, L. Chen, C. Ledig, D. Rueckert, and B. Glocker. Multi-scale 3d convolutional neural networks for lesion segmentation in brain mri. 2015.

G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu. Lightgbm: A highly efficient gradient boosting decision tree. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3146–3154. Curran Associates, Inc., 2017.

D. Kingma and J. Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

D. Kingma and M. Welling. Auto-encoding variational bayes. 12 2014.

A. Knebe, S. R. Knollmann, S. I. Muldrew, F. R. Pearce, M. A. Aragon-Calvo, Y. Ascasibar, P. S. Behroozi, D. Ceverino, S. Colombi, J. Diemand, K. Dolag, B. L. Falck, P. Fasel, J. Gardner, S. Gottlöber, C.-H. Hsu, F. Iannuzzi, A. Klypin, Z. Lukić, M. Maciejewski, C. McBride, M. C. Neyrinck, S. Planelles, D. Potter, V. Quilis, Y. Rasera, J. I. Read, P. M. Ricker, F. Roy, V. Springel, J. Stadel, G. Stinson, P. M. Sutter, V. Turchaninov, D. Tweed, G. Yepes, and M. Zemp. Haloes gone MAD: The Halo-Finder Comparison Project. MNRAS, 415:2293–2318, Aug. 2011. doi: 10.1111/j.1365-2966.2011.18858.x.

A. Knebe, F. R. Pearce, H. Lux, Y. Ascasibar, P. Behroozi, J. Casado, C. C. Moran, J. Diemand, K. Dolag, R. Dominguez-Tenreiro, P. Elahi, B. Falck, S. Gottlöber, J. Han, A. Klypin, Z. Lukić, M. Maciejewski, C. K. McBride, M. E. Merchán, S. I. Muldrew, M. Neyrinck, J. Onions, S. Planelles, D. Potter, V. Quilis, Y. Rasera, P. M. Ricker, F. Roy, A. N. Ruiz, M. A. Sgró, V. Springel, J. Stadel, P. M. Sutter, D. Tweed, and M. Zemp. Structure finding in cosmological simulations: the state of affairs. MNRAS, 435:1618–1658, Oct. 2013. doi: 10.1093/mnras/stt1403.

S. R. Knollmann and A. Knebe. AHF: Amiga's Halo Finder. ApJS, 182:608–624, June 2009. doi: 10.1088/0067-0049/182/2/608.

D. Kodi Ramanah, T. Charnock, and G. Lavaux. Painting halos from cosmic density fields of dark matter with physically motivated neural networks. Phys. Rev. D, 100(4):043515, Aug 2019. doi: 10.1103/PhysRevD.100.043515.

R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'95, pages 1137–1143, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1-55860-363-8.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. URL http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

M. Kuhlen, M. Vogelsberger, and R. Angulo. Numerical simulations of the dark universe: State of the art and the next decade. *Physics of the Dark Universe*, 1:50–93, Nov. 2012.

M. Kuhn and K. Johnson. Applied predictive modeling. 2013.

S. Kullback and R. A. Leibler. On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86, 03 1951. doi: 10.1214/aoms/1177729694. URL https://doi.org/10.1214/aoms/1177729694.

O. Lahav, P. B. Lilje, J. R. Primack, and M. J. Rees. Dynamical effects of the cosmological constant. MNRAS, 251:128–136, Jul 1991. doi: 10.1093/mnras/251.1.128.

O. Lahav, A. Naim, R. J. Buta, H. G. Corwin, G. de Vaucouleurs, A. Dressler, J. P. Huchra, S. van den Bergh, S. Raychaudhury, L. Sodre, and et al. Galaxies, human eyes, and artificial neural networks. *Science*, 267(5199):859862, Feb 1995. ISSN 1095-9203. doi: 10.1126/science.267.5199.859. URL http://dx.doi.org/10.1126/science.267.5199.859.

O. Lahav, A. Naim, J. Sodré, L., and M. C. Storrie-Lombardi. Neural computation as a tool for galaxy classification: methods and examples. MNRAS, 283:207, Nov 1996. doi: 10.1093/mnras/283.1.207.

Q. V. Le, M. Ranzato, R. Monga, M. Devin, G. S. Corrado, K. Chen, J. Dean, and A. Y. Ng. Building high-level features using large scale unsupervised learning. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8595–8598, 2011.

F. Leclercq, J. Jasche, and B. Wandelt. Cosmic web-type classification using decision theory. A&A, 576:L17, Apr 2015. doi: 10.1051/0004-6361/201526006.

Y. Lecun and Y. Bengio. *Convolutional networks for images, speech, and time-series*. MIT Press, 1995.

Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 5 2015. ISSN 0028-0836. doi: 10.1038/nature14539.

A. Lewis, A. Challinor, and A. Lasenby. Efficient Computation of Cosmic Microwave Background Anisotropies in Closed Friedmann-Robertson-Walker Models. ApJ, 538(2):473–476, Aug 2000. doi: 10.1086/309179.

J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu. Feature selection: A data perspective. *CoRR*, abs/1601.07996, 2016.

C. C. Lin, L. Mestel, and F. H. Shu. The Gravitational Collapse of a Uniform Spheroid. ApJ, 142: 1431, Nov 1965. doi: 10.1086/148428.

A. Linde. Scalar field fluctuations in the expanding universe and the new inflationary universe scenario. *Physics Letters B*, 116(5):335 – 339, 1982. ISSN 0370-2693. doi: https://doi.org/10.1016/0370-2693(82)90293-3. URL http://www.sciencedirect.com/science/article/pii/0370269382902933.

A. D. Linde. A new inflationary universe scenario: A possible solution of the horizon, flatness, homogeneity, isotropy and primordial monopole problems. *Physics Letters B*, 108:389–393, Feb. 1982. doi: 10.1016/0370-2693(82)91219-9.

M. Lochner, J. D. McEwen, H. V. Peiris, O. Lahav, and M. K. Winter. Photometric Supernova Classification with Machine Learning. ApJS, 225:31, Aug. 2016. doi: 10.3847/0067-0049/225/2/31.

G. Louppe, L. Wehenkel, A. Sutera, and P. Geurts. Understanding variable importances in forests of randomized trees. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 431–439. Curran Associates, Inc., 2013. URL http://papers.nips.cc/paper/4928-understanding-variable-importances-in-forests-of-randomized-trees.pdf.

L. Lucie-Smith, H. V. Peiris, A. Pontzen, and M. Lochner. Machine learning cosmological structure formation. MNRAS, 479:3405–3414, Sept. 2018. doi: 10.1093/mnras/sty1719.

L. Lucie-Smith, H. V. Peiris, and A. Pontzen. An interpretable machine-learning framework for dark matter halo formation. MNRAS, 490(1):331–342, Nov 2019. doi: 10.1093/mnras/stz2599.

C.-P. Ma and E. Bertschinger. Cosmological Perturbation Theory in the Synchronous and Conformal Newtonian Gauges. ApJ, 455:7, Dec. 1995. doi: 10.1086/176550.

D. J. C. MacKay. Information theory, inference, and learning algorithms. *IEEE Transactions on Information Theory*, 50:2544–2545, 2003.

S. J. Maddox, W. J. Sutherland, G. Efstathiou, and J. Loveday. The APM galaxy survey. I - APM measurements and star-galaxy separation. MNRAS, 243:692–712, Apr. 1990.

P. H. Maehoenen and P. J. Hakala. Automated Source Classification Using a Kohonen Network. ApJ, 452:L77, Oct 1995. doi: 10.1086/309697.

M. Maggiore and A. Riotto. The Halo Mass Function from Excursion Set Theory. I. Gaussian Fluctuations with Non-Markovian Dependence on the Smoothing Scale. ApJ, 711:907–927, Mar. 2010. doi: 10.1088/0004-637X/711/2/907.

J. Martin. Everything you always wanted to know about the cosmological constant problem (but were afraid to ask). *Comptes Rendus Physique*, 13(6-7):566–665, Jul 2012. doi: 10.1016/j.crhy.2012.04.008.

R. Massey, T. Kitching, and J. Richard. The dark matter of gravitational lensing. *Reports on Progress in Physics*, 73(8):086901, Aug 2010. doi: 10.1088/0034-4885/73/8/086901.

J. C. Mather, E. S. Cheng, D. A. Cottingham, R. E. Eplee, Jr., D. J. Fixsen, T. Hewagama, R. B. Isaacman, K. A. Jensen, S. S. Meyer, P. D. Noerdlinger, S. M. Read, L. P. Rosen, R. A. Shafer, E. L. Wright, C. L. Bennett, N. W. Boggess, M. G. Hauser, T. Kelsall, S. H. Moseley, Jr., R. F. Silverberg, G. F. Smoot, R. Weiss, and D. T. Wilkinson. Measurement of the cosmic microwave background spectrum by the COBE FIRAS instrument. ApJ, 420:439–444, Jan. 1994. doi: 10.1086/173574.

A. Mathuriya, D. Bard, P. Mendygral, L. Meadows, J. Arnemann, L. Shao, S. He, T. Kärnä, D. Moise, S. J. Pennycook, K. Maschhoff, J. Sewall, N. Kumar, S. Ho, M. F. Ringenburg, Prabhat, and V. Lee. Cosmoflow: Using deep learning to learn the universe at scale. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis*, SC '18, pages 65:1–65:11, Piscataway, NJ, USA, 2018. IEEE Press. doi: 10.1109/SC.2018.00068. URL https://doi.org/10.1109/SC.2018.00068.

M. McLeod, N. Libeskind, O. Lahav, and Y. Hoffman. Estimating the mass of the Local Group using machine learning applied to numerical simulations. J. Cosmology Astropart. Phys., 2017(12): 034, Dec 2017. doi: 10.1088/1475-7516/2017/12/034.

P. Mehta, M. Bukov, C.-H. Wang, A. r. G. R. Day, C. Richardson, C. K. Fisher, and D. J. Schwab. A high-bias, low-variance introduction to Machine Learning for physicists. Phys. Rep., 810:1–124, May 2019. doi: 10.1016/j.physrep.2019.03.001.

J. Merten, C. Giocoli, M. Baldi, M. Meneghetti, A. Peel, F. Lalande, J.-L. Starck, and V. Pettorino. On the dissection of degenerate cosmologies with machine learning. MNRAS, 487(1):104–122, Jul 2019. doi: 10.1093/mnras/stz972.

H. Mo, F. C. van den Bosch, and S. White. *Galaxy Formation and Evolution*. 2010.

C. Modi, Y. Feng, and U. Seljak. Cosmological reconstruction from galaxy light: neural network based light-matter connection. J. Cosmology Astropart. Phys., 2018(10):028, Oct 2018. doi: 10.1088/1475-7516/2018/10/028.

A. Moss. Improved Photometric Classification of Supernovae using Deep Learning. *arXiv e-prints*, art. arXiv:1810.06441, Oct 2018.

V. F. Mukhanov and G. V. Chibisov. Quantum Fluctuations and a Nonsingular Universe. *JETP Lett.*, 33:532–535, 1981. [Pisma Zh. Eksp. Teor. Fiz.33,549(1981)].

S. I. Muldrew, F. R. Pearce, and C. Power. The accuracy of subhalo detection. MNRAS, 410: 2617–2624, Feb. 2011. doi: 10.1111/j.1365-2966.2010.17636.x.

K. P. Murphy. *Machine learning: a probabilistic perspective*. Cambridge, MA, 2012.

E. O. Nadler, Y.-Y. Mao, R. H. Wechsler, S. Garrison-Kimmel, and A. Wetzel. Modeling the Impact of Baryons on Subhalo Populations with Machine Learning. ApJ, 859(2):129, Jun 2018. doi: 10.3847/1538-4357/aac266.

A. Naim, O. Lahav, L. Sodre, and M. C. Storrie-Lombardi. Automated morphological classification of apm galaxies by supervised artificial neural networks. *Monthly Notices of the Royal Astronomical Society*, 275(3):567590, Aug 1995. ISSN 1365-2966. doi: 10.1093/mnras/275.3.567. URL http://dx.doi.org/10.1093/mnras/275.3.567.

V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pages 807–814, USA, 2010. Omnipress. ISBN 978-1-60558-907-7. URL http://dl.acm.org/citation.cfm?id=3104322.3104425.

A. Niculescu-Mizil and R. Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 625–632, New York, NY, USA, 2005. ACM. ISBN 1-59593-180-5. doi: 10.1145/1102351.1102430. URL http://doi.acm.org/10.1145/1102351.1102430.

M. A. Nielsen. *Neural networks and deep learning*, volume 25. Determination press San Francisco, CA, USA:, 2015.

M. Ntampaka, H. Trac, D. J. Sutherland, N. Battaglia, B. Póczos, and J. Schneider. A MACHINE LEARNING APPROACH FOR DYNAMICAL MASS MEASUREMENTS OF GALAXY CLUSTERS. *The Astrophysical Journal*, 803(2):50, apr 2015. doi: 10.1088/0004-637x/803/2/50. URL https://doi.org/10.1088%2F0004-637x%2F803%2F2%2F50.

M. Ntampaka, C. Avestruz, S. Boada, J. Caldeira, J. Cisewski-Kehe, R. Di Stefano, C. Dvorkin, A. E. Evrard, A. Farahi, D. Finkbeiner, S. Genel, A. Goodman, A. Goulding, S. Ho, A. Kosowsky, P. La Plante, F. Lanusse, M. Lochner, R. Mandelbaum, D. Nagai, J. A. Newman, B. Nord, J. E. G. Peek, A. Peel, B. Poczos, M. M. Rau, A. Siemiginowska, D. J. Sutherland, H. Trac, and B. Wandelt. The Role of Machine Learning in the Next Decade of Cosmology. BAAS, 51(3):14, May 2019.

C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall. Activation functions: Comparison of trends in practice and research for deep learning. arxiv 2018. *arXiv preprint arXiv:1811.03378*, 2018.

S. C. Odewahn, E. B. Stockwell, R. L. Pennington, R. M. Humphreys, and W. A. Zumach. Automated Star/Galaxy Discrimination With Neural Networks. AJ, 103:318, Jan 1992. doi: 10.1086/116063.

A. Oka, S. Saito, T. Nishimichi, A. Taruya, and K. Yamamoto. Simultaneous constraints on the growth of structure and cosmic expansion from the multipole power spectra of the SDSS DR7 LRG sample. MNRAS, 439(3):2515–2530, Apr 2014. doi: 10.1093/mnras/stu111.

C. Olah, A. Mordvintsev, and L. Schubert. Feature visualization. *Distill*, 2, 11 2017. doi: 10.23915/distill.00007.

C. Olah, A. Satyanarayan, I. Johnson, S. Carter, L. Schubert, K. Ye, and A. Mordvintsev. The building blocks of interpretability. *Distill*, 2018. URL https://distill.pub/2018/building-blocks/.

J. H. Oort. The force exerted by the stellar system in the direction perpendicular to the galactic plane and some related problems. Bull. Astron. Inst. Netherlands, 6:249, Aug. 1932.

J. P. Ostriker and P. J. Steinhardt. Cosmic Concordance. *arXiv e-prints*, art. astro-ph/9505066, May 1995.

W. Ouyang, X. Zeng, X. Wang, S. Qiu, P. Luo, Y. Tian, H. Li, S. Yang, Z. Wang, H. Li, et al. Deepid-net: Object detection with deformable part based convolutional neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(7):1320–1334, 2016.

S. Pan, M. Liu, J. Forero-Romero, C. G. Sabiu, Z. Li, H. Miao, and X.-D. Li. Cosmological parameter estimation from large-scale structure deep learning. *arXiv e-prints*, art. arXiv:1908.10590, Aug 2019.

A. Paranjape and R. K. Sheth. Peaks theory and the excursion set approach. *Monthly Notices of the Royal Astronomical Society*, 426(4):2789–2796, 2012. doi: 10.1111/j.1365-2966.2012.21911.x. URL +http://dx.doi.org/10.1111/j.1365-2966.2012.21911.x.

R. D. Peccei and H. R. Quinn. Cp conservation in the presence of instantons. *Phys. Rev. Lett.*, 38 (ITP-568-STANFORD):1440–1443, 1977.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

P. Peebles. *The Large-scale Structure of the Universe*. Princeton Series in Physics. Princeton University Press, 1980. ISBN 9780691082400. URL https://books.google.co.uk/books?id=0_BPaHFtX1YC.

A. Peel, F. Lalande, J.-L. Starck, V. Pettorino, J. Merten, C. Giocoli, M. Meneghetti, and M. Baldi. Distinguishing standard and modified gravity cosmologies with machine learning. Phys. Rev. D, 100(2):023508, Jul 2019. doi: 10.1103/PhysRevD.100.023508.

A. A. Penzias and R. W. Wilson. A Measurement of excess antenna temperature at 4080-Mc/s. *Astrophys. J.*, 142:419–421, 1965. doi: 10.1086/148307.

W. J. Percival, C. M. Baugh, J. Bland-Hawthorn, T. Bridges, R. Cannon, S. Cole, M. Colless, C. Collins, W. Couch, G. Dalton, R. De Propris, S. P. Driver, G. Efstathiou, R. S. Ellis, C. S. Frenk, K. Glazebrook, C. Jackson, O. Lahav, I. Lewis, S. Lumsden, S. Maddox, S. Moody, P. Norberg, J. A. Peacock, B. A. Peterson, W. Sutherland, and K. Taylor. The 2dF Galaxy Redshift Survey: the power spectrum and the matter content of the Universe. MNRAS, 327:1297–1306, Nov. 2001. doi: 10.1046/j.1365-8711.2001.04827.x.

S. Perlmutter, G. Aldering, G. Goldhaber, R. A. Knop, P. Nugent, P. G. Castro, S. Deustua, S. Fabbro, A. Goobar, D. E. Groom, I. M. Hook, A. G. Kim, M. Y. Kim, J. C. Lee, N. J. Nunes, R. Pain, C. R. Pennypacker, R. Quimby, C. Lidman, R. S. Ellis, M. Irwin, R. G. McMahon, P. Ruiz-Lapuente, N. Walton, B. Schaefer, B. J. Boyle, A. V. Filippenko, T. Matheson, A. S. Fruchter, N. Panagia, H. J. M. Newberg, W. J. Couch, and T. S. C. Project. Measurements of Ω and Λ from 42 High-Redshift Supernovae. ApJ, 517(2):565–586, Jun 1999. doi: 10.1086/307221.

Planck Collaboration, P. A. R. Ade, N. Aghanim, M. Arnaud, M. Ashdown, J. Aumont, C. Baccigalupi, A. J. Banday, R. B. Barreiro, J. G. Bartlett, N. Bartolo, E. Battaner, R. Battye, K. Benabed, A. Benoît, A. Benoit-Lévy, J. P. Bernard, M. Bersanelli, P. Bielewicz, J. J. Bock, A. Bonaldi, L. Bonavera, J. R. Bond, J. Borrill, F. R. Bouchet, F. Boulanger, M. Bucher, C. Burigana, R. C. Butler, E. Calabrese, J. F. Cardoso, A. Catalano, A. Challinor, A. Chamballu, R. R. Chary, H. C. Chiang, J. Chluba, P. R. Christensen, S. Church, D. L. Clements, S. Colombi, L. P. L. Colombo, C. Combet, A. Coulais, B. P. Crill, A. Curto, F. Cuttaia, L. Danese, R. D. Davies, R. J. Davis, P. de Bernardis, A. de Rosa, G. de Zotti, J. Delabrouille, F. X. Désert, E. Di Valentino, C. Dickinson, J. M. Diego, K. Dolag, H. Dole, S. Donzelli, O. Doré, M. Douspis, A. Ducout, J. Dunkley, X. Dupac, G. Efstathiou, F. Elsner, T. A. Enßlin, H. K. Eriksen, M. Farhang, J. Fergusson, F. Finelli, O. Forni, M. Frailis, A. A. Fraisse, E. Franceschi, A. Frejsel, S. Galeotta, S. Galli, K. Ganga, C. Gauthier, M. Gerbino, T. Ghosh, M. Giard, Y. Giraud-Héraud, E. Giusarma, E. Gjerløw, J. González-Nuevo, K. M. Górski, S. Gratton, A. Gregorio, A. Gruppuso, J. E. Gudmundsson, J. Hamann, F. K. Hansen, D. Hanson, D. L. Harrison, G. Helou, S. Henrot-Versillé, C. Hernández-Monteagudo, D. Herranz, S. R. Hildebrand t, E. Hivon, M. Hobson, W. A. Holmes, A. Hornstrup, W. Hovest, Z. Huang, K. M. Huffenberger, G. Hurier, A. H. Jaffe, T. R. Jaffe, W. C. Jones, M. Juvela, E. Keihänen, R. Keskitalo, T. S. Kisner, R. Kneissl, J. Knoche, L. Knox, M. Kunz, H. Kurki-Suonio, G. Lagache, A. Lähteenmäki, J. M. Lamarre, A. Lasenby, M. Lattanzi, C. R. Lawrence, J. P. Leahy, R. Leonardi, J. Lesgourgues, F. Levrier, A. Lewis, M. Liguori, P. B. Lilje, M. Linden-Vørnle, M. López-Caniego, P. M. Lubin, J. F. Macías-Pérez, G. Maggio, D. Maino, N. Mandolesi, A. Mangilli, A. Marchini, M. Maris, P. G. Martin, M. Martinelli, E. Martínez-González, S. Masi, S. Matarrese, P. McGehee, P. R. Meinhold, A. Melchiorri, J. B. Melin, L. Mendes, A. Mennella, M. Migliaccio, M. Millea, S. Mitra, M. A. Miville-Deschênes, A. Moneti, L. Montier, G. Morgante, D. Mortlock, A. Moss, D. Munshi, J. A. Murphy, P. Naselsky, F. Nati, P. Natoli, C. B. Netterfield, H. U. Nørgaard-Nielsen, F. Noviello, D. Novikov, I. Novikov, C. A. Oxborrow, F. Paci, L. Pagano, F. Pajot, R. Paladini, D. Paoletti, B. Partridge, F. Pasian, G. Patanchon, T. J. Pearson, O. Perdereau, L. Perotto, F. Perrotta, V. Pettorino, F. Piacentini, M. Piat, E. Pierpaoli, D. Pietrobon, S. Plaszczynski, E. Pointecouteau, G. Polenta, L. Popa, G. W. Pratt, G. Prézeau, S. Prunet, J. L. Puget, J. P. Rachen, W. T. Reach, R. Rebolo, M. Reinecke, M. Remazeilles, C. Renault, A. Renzi, I. Ristorcelli, G. Rocha, C. Rosset, M. Rossetti, G. Roudier, B. Rouillé d'Orfeuil, M. Rowan-Robinson, J. A. Rubiño-Martín, B. Rusholme, N. Said, V. Salvatelli, L. Salvati, M. Sandri, D. Santos, M. Savelainen, G. Savini, D. Scott, M. D. Seiffert, P. Serra, E. P. S. Shellard, L. D. Spencer, M. Spinelli, V. Stolyarov, R. Stompor, R. Sudiwala, R. Sunyaev, D. Sutton, A. S. Suur-Uski, J. F. Sygnet, J. A. Tauber, L. Terenzi, L. Toffolatti, M. Tomasi, M. Tristram, T. Trombetti, M. Tucci, J. Tuovinen, M. Türler, G. Umana, L. Valenziano, J. Valiviita, F. Van Tent, P. Vielva, F. Villa, L. A. Wade, B. D. Wandelt, I. K. Wehus, M. White, S. D. M. White, A. Wilkinson, D. Yvon, A. Zacchei, and A. Zonca. Planck 2015 results. XIII. Cosmological parameters. A&A, 594:A13, Sep 2016. doi: 10.1051/0004-6361/201525830.

Planck Collaboration, N. Aghanim, Y. Akrami, M. Ashdown, J. Aumont, C. Baccigalupi, M. Ballardini, A. J. Banday, R. B. Barreiro, N. Bartolo, S. Basak, R. Battye, K. Benabed, J.-P. Bernard, M. Bersanelli, P. Bielewicz, J. J. Bock, J. R. Bond, J. Borrill, F. R. Bouchet, F. Boulanger, M. Bucher, C. Burigana, R. C. Butler, E. Calabrese, J.-F. Cardoso, J. Carron, A. Challinor, H. C. Chiang, J. Chluba, L. P. L. Colombo, C. Combet, D. Contreras, B. P. Crill, F. Cuttaia, P. de Bernardis, G. de Zotti, J. Delabrouille, J.-M. Delouis, E. Di Valentino, J. M. Diego, O. Doré, M. Douspis, A. Ducout, X. Dupac, S. Dusini, G. Efstathiou, F. Elsner, T. A. Enßlin, H. K. Eriksen, Y. Fantaye, M. Farhang, J. Fergusson, R. Fernandez-Cobos, F. Finelli, F. Forastieri, M. Frailis, E. Franceschi, A. Frolov, S. Galeotta, S. Galli, K. Ganga, R. T. Génova-Santos, M. Gerbino, T. Ghosh, J. González-Nuevo, K. M. Górski, S. Gratton, A. Gruppuso, J. E. Gudmundsson, J. Hamann, W. Handley, D. Herranz, E. Hivon, Z. Huang, A. H. Jaffe, W. C. Jones, A. Karakci, E. Keihänen, R. Keskitalo, K. Kiiveri, J. Kim, T. S. Kisner, L. Knox, N. Krachmalnicoff, M. Kunz, H. Kurki-Suonio, G. Lagache, J.-M. Lamarre, A. Lasenby, M. Lattanzi, C. R. Lawrence, M. Le Jeune, P. Lemos, J. Lesgourgues, F. Levrier, A. Lewis, M. Liguori, P. B. Lilje, M. Lilley, V. Lindholm, M. López-Caniego, P. M. Lubin, Y.-Z. Ma, J. F. Macías-Pérez, G. Maggio, D. Maino, N. Mandolesi, A. Mangilli, A. Marcos-Caballero, M. Maris,

P. G. Martin, M. Martinelli, E. Martínez-González, S. Matarrese, N. Mauri, J. D. McEwen, P. R. Meinhold, A. Melchiorri, A. Mennella, M. Migliaccio, M. Millea, S. Mitra, M.-A. Miville-Deschênes, D. Molinari, L. Montier, G. Morgante, A. Moss, P. Natoli, H. U. Nørgaard-Nielsen, L. Pagano, D. Paoletti, B. Partridge, G. Patanchon, H. V. Peiris, F. Perrotta, V. Pettorino, F. Piacentini, L. Polastri, G. Polenta, J.-L. Puget, J. P. Rachen, M. Reinecke, M. Remazeilles, A. Renzi, G. Rocha, C. Rosset, G. Roudier, J. A. Rubiño-Martín, B. Ruiz-Granados, L. Salvati, M. Sandri, M. Savelainen, D. Scott, E. P. S. Shellard, C. Sirignano, G. Sirri, L. D. Spencer, R. Sunyaev, A.-S. Suur-Uski, J. A. Tauber, D. Tavagnacco, M. Tenti, L. Toffolatti, M. Tomasi, T. Trombetti, L. Valenziano, J. Valiviita, B. Van Tent, L. Vibert, P. Vielva, F. Villa, N. Vittorio, B. D. Wandelt, I. K. Wehus, M. White, S. D. M. White, A. Zacchei, and A. Zonca. Planck 2018 results. VI. Cosmological parameters. *preprint (arxiv: 1807.06209)*, July 2018a.

Planck Collaboration, Y. Akrami, F. Arroja, M. Ashdown, J. Aumont, C. Baccigalupi, M. Ballardini, A. J. Banday, R. B. Barreiro, N. Bartolo, S. Basak, R. Battye, K. Benabed, J. P. Bernard, M. Bersanelli, P. Bielewicz, J. J. Bock, J. R. Bond, J. Borrill, F. R. Bouchet, F. Boulanger, M. Bucher, C. Burigana, R. C. Butler, E. Calabrese, J. F. Cardoso, J. Carron, B. Casaponsa, A. Challinor, H. C. Chiang, L. P. L. Colombo, C. Combet, D. Contreras, B. P. Crill, F. Cuttaia, P. de Bernardis, G. de Zotti, J. Delabrouille, J. M. Delouis, F. X. Désert, E. Di Valentino, C. Dickinson, J. M. Diego, S. Donzelli, O. Doré, M. Douspis, A. Ducout, X. Dupac, G. Efstathiou, F. Elsner, T. A. Enßlin, H. K. Eriksen, E. Falgarone, Y. Fantaye, J. Fergusson, R. Fernandez-Cobos, F. Finelli, F. Forastieri, M. Frailis, E. Franceschi, A. Frolov, S. Galeotta, S. Galli, K. Ganga, R. T. Génova-Santos, M. Gerbino, T. Ghosh, J. González-Nuevo, K. M. Górski, S. Gratton, A. Gruppuso, J. E. Gudmundsson, J. Hamann, W. Hand ley, F. K. Hansen, G. Helou, D. Herranz, E. Hivon, Z. Huang, A. H. Jaffe, W. C. Jones, A. Karakci, E. Keihänen, R. Keskitalo, K. Kiiveri, J. Kim, T. S. Kisner, L. Knox, N. Krachmalnicoff, M. Kunz, H. Kurki-Suonio, G. Lagache, J. M. Lamarre, M. Langer, A. Lasenby, M. Lattanzi, C. R. Lawrence, M. Le Jeune, J. P. Leahy, J. Lesgourgues, F. Levrier, A. Lewis, M. Liguori, P. B. Lilje, M. Lilley, V. Lindholm, M. López-Caniego, P. M. Lubin, Y. Z. Ma, J. F. Macías-Pérez, G. Maggio, D. Maino, N. Mand olesi, A. Mangilli, A. Marcos-Caballero, M. Maris, P. G. Martin, E. Martínez-González, S. Matarrese, N. Mauri, J. D. McEwen, P. D. Meerburg, P. R. Meinhold, A. Melchiorri, A. Mennella, M. Migliaccio, M. Millea, S. Mitra, M. A. Miville-Deschênes, D. Molinari, A. Moneti, L. Montier, G. Morgante, A. Moss, S. Mottet, M. Münchmeyer, P. Natoli, H. U. Nørgaard-Nielsen, C. A. Oxborrow, L. Pagano, D. Paoletti, B. Partridge, G. Patanchon, T. J. Pearson, M. Peel, H. V. Peiris, F. Perrotta, V. Pettorino, F. Piacentini, L. Polastri, G. Polenta, J. L. Puget, J. P. Rachen, M. Reinecke, M. Remazeilles, A. Renzi, G. Rocha, C. Rosset, G. Roudier, J. A. Rubiño-Martín, B. Ruiz-Granados, L. Salvati, M. Sandri, M. Savelainen, D. Scott, E. P. S. Shellard, M. Shiraishi, C. Sirignano, G. Sirri, L. D. Spencer, R. Sunyaev, A. S. Suur-Uski, J. A. Tauber, D. Tavagnacco, M. Tenti, L. Terenzi, L. Toffolatti, M. Tomasi, T. Trombetti, J. Valiviita, B. Van Tent, L. Vibert, P. Vielva, F. Villa, N. Vittorio, B. D. Wandelt, I. K. Wehus, M. White, S. D. M. White, A. Zacchei, and A. Zonca. Planck 2018 results. I. Overview and the cosmological legacy of Planck. *arXiv e-prints*, art. arXiv:1807.06205, Jul 2018b.

A. Pontzen and F. Governato. How supernova feedback turns dark matter cusps into cores. MNRAS, 421:3464–3471, Apr. 2012. doi: 10.1111/j.1365-2966.2012.20571.x.

A. Pontzen, R. Roškar, G. S. Stinson, R. Woods, D. M. Reed, J. Coles, and T. R. Quinn. pynbody: Astrophysics Simulation Analysis for Python, 2013. Astrophysics Source Code Library, ascl:1305.002.

W. H. Press and P. Schechter. Formation of Galaxies and Clusters of Galaxies by Self-Similar Gravitational Condensation. ApJ, 187:425–438, Feb. 1974. doi: 10.1086/152650.

J. R. Quinlan. Induction of decision trees. *Mach. Learn.*, 1(1):81–106, Mar. 1986. ISSN 0885-6125. doi: 10.1023/A:1022643204877. URL `http://dx.doi.org/10.1023/A:1022643204877`.

R. Rao and G. Fung. On the dangers of cross-validation. an experimental evaluation. pages 588–596, 04 2008. doi: 10.1137/1.9781611972788.54.

S. Ravanbakhsh, J. Oliva, S. Fromenteau, L. C. Price, S. Ho, J. Schneider, and B. Póczos. Estimating cosmological parameters from the dark matter distribution. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 2407–2416. JMLR.org, 2016. URL http://dl.acm.org/citation.cfm?id=3045390.3045644.

D. Reed, J. Gardner, T. Quinn, J. Stadel, M. Fardal, G. Lake, and F. Governato. Evolution of the mass function of dark matter haloes. MNRAS, 346:565–572, Dec. 2003. doi: 10.1046/j.1365-2966.2003.07113.x.

D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In E. P. Xing and T. Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1278–1286, Bejing, China, 22–24 Jun 2014. PMLR. URL http://proceedings.mlr.press/v32/rezende14.html.

A. G. Riess, A. V. Filippenko, P. Challis, A. Clocchiatti, A. Diercks, P. M. Garnavich, R. L. Gilliland, C. J. Hogan, S. Jha, R. P. Kirshner, B. Leibundgut, M. M. Phillips, D. Reiss, B. P. Schmidt, R. A. Schommer, R. C. Smith, J. Spyromilio, C. Stubbs, N. B. Suntzeff, and J. Tonry. Observational Evidence from Supernovae for an Accelerating Universe and a Cosmological Constant. AJ, 116:1009–1038, Sept. 1998. doi: 10.1086/300499.

A. G. Riess, S. Casertano, W. Yuan, L. M. Macri, and D. Scolnic. Large Magellanic Cloud Cepheid Standards Provide a 1% Foundation for the Determination of the Hubble Constant and Stronger Evidence for Physics beyond ΛCDM. ApJ, 876(1):85, May 2019. doi: 10.3847/1538-4357/ab1422.

R. Roscher, B. Bohn, M. F. Duarte, and J. Garcke. Explainable Machine Learning for Scientific Insights and Discoveries. *arXiv e-prints*, art. arXiv:1905.08883, May 2019.

M. Rosenblatt. Remarks on some nonparametric estimates of a density function. *Ann. Math. Statist.*, 27(3):832–837, 09 1956. doi: 10.1214/aoms/1177728190. URL https://doi.org/10.1214/aoms/1177728190.

V. C. Rubin and W. K. Ford, Jr. Rotation of the Andromeda Nebula from a Spectroscopic Survey of Emission Regions. ApJ, 159:379, Feb. 1970. doi: 10.1086/150317.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986. doi: 10.1038/323533a0. URL https://doi.org/10.1038/323533a0.

R. Salakhutdinov and G. Hinton. Deep boltzmann machines. In *Artificial intelligence and statistics*, pages 448–455, 2009.

S. L. Salzberg. C4.5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993. *Machine Learning*, 16:235–240, 1994.

S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry. How does batch normalization help optimization? In *Proceedings of the 32Nd International Conference on Neural Information Processing Systems*, NIPS'18, pages 2488–2498, USA, 2018. Curran Associates Inc. URL http://dl.acm.org/citation.cfm?id=3327144.3327174.

G. Sato-Polito, E. D. Kovetz, and M. Kamionkowski. Constraints on the primordial curvature power spectrum from primordial black holes. Phys. Rev. D, 100(6):063521, Sep 2019. doi: 10.1103/PhysRevD.100.063521.

R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee. Boosting the margin: a new explanation for the effectiveness of voting methods. *Ann. Statist.*, 26(5):1651–1686, 10 1998. doi: 10.1214/aos/1024691352. URL https://doi.org/10.1214/aos/1024691352.

J. Schmelzle, A. Lucchi, T. Kacprzak, A. Amara, R. Sgier, A. Réfrégier, and T. Hofmann. Cosmological model discrimination with Deep Learning. *arXiv e-prints*, art. arXiv:1707.05167, Jul 2017.

P. Schneider. *Extragalactic Astronomy and Cosmology*. Springer, 2006.

M. Seldner, B. Siebers, E. J. Groth, and P. J. E. Peebles. Charge States of Low Energy Ions from the Sun. AJ, 82:249, Apr 1977. doi: 10.1086/112039.

R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2016.

R. K. Sheth and G. Tormen. Large-scale bias and the peak background split. MNRAS, 308:119–126, Sept. 1999. doi: 10.1046/j.1365-8711.1999.02692.x.

R. K. Sheth, H. J. Mo, and G. Tormen. Ellipsoidal collapse and an improved model for the number and spatial distribution of dark matter haloes. MNRAS, 323:1–12, May 2001. doi: 10.1046/j.1365-8711.2001.04006.x.

D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–503, 2016. URL `http://www.nature.com/nature/journal/v529/n7587/full/nature16961.html`.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, abs/1312.6034, 2013.

V. M. Slipher. Spectrographic Observations of Nebulae. *Popular Astronomy*, 23:21–24, Jan. 1915.

G. F. Smoot, C. L. Bennett, A. Kogut, E. L. Wright, J. Aymon, N. W. Boggess, E. S. Cheng, G. de Amici, S. Gulkis, M. G. Hauser, G. Hinshaw, P. D. Jackson, M. Janssen, E. Kaita, T. Kelsall, P. Keegstra, C. Lineweaver, K. Loewenstein, P. Lubin, J. Mather, S. S. Meyer, S. H. Moseley, T. Murdock, L. Rokke, R. F. Silverberg, L. Tenorio, R. Weiss, and D. T. Wilkinson. Structure in the COBE differential microwave radiometer first-year maps. ApJ, 396:L1–L5, Sept. 1992. doi: 10.1086/186504.

M. Soares-Santos et al. First Measurement of the Hubble Constant from a Dark Standard Siren using the Dark Energy Survey Galaxies and the LIGO/Virgo BinaryBlack-hole Merger GW170814. *Astrophys. J.*, 876(1):L7, 2019. doi: 10.3847/2041-8213/ab14f1.

V. Springel. The cosmological simulation code GADGET-2. MNRAS, 364:1105–1134, Dec. 2005. doi: 10.1111/j.1365-2966.2005.09655.x.

V. Springel, N. Yoshida, and S. D. M. White. GADGET: a code for collisionless and gasdynamical cosmological simulations. New A, 6:79–117, Apr. 2001. doi: 10.1016/S1384-1076(01)00042-2.

V. Springel, S. D. M. White, A. Jenkins, C. S. Frenk, N. Yoshida, L. Gao, J. Navarro, R. Thacker, D. Croton, J. Helly, J. A. Peacock, S. Cole, P. Thomas, H. Couchman, A. Evrard, J. Colberg, and F. Pearce. Simulations of the formation, evolution and clustering of galaxies and quasars. Nature, 435:629–636, June 2005. doi: 10.1038/nature03597.

V. Springel, C. S. Frenk, and S. D. M. White. The large-scale structure of the Universe. Nature, 440 (7088):1137–1144, Apr 2006. doi: 10.1038/nature04805.

J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. *CoRR*, abs/1412.6806, 2014.

A. A. Starobinsky. Dynamics of Phase Transition in the New Inflationary Universe Scenario and Generation of Perturbations. *Phys. Lett.*, 117B:175–178, 1982. doi: 10.1016/0370-2693(82)90541-X.

G. Steigman and M. S. Turner. Cosmological constraints on the properties of weakly interacting massive particles. *Nuclear Physics B*, 253:375 – 386, 1985. ISSN 0550-3213. doi: https://doi.org/10.1016/0550-3213(85)90537-1. URL http://www.sciencedirect.com/science/article/pii/0550321385905371.

M. C. Storrie-Lombardi, O. Lahav, J. Sodre, L., and L. J. Storrie-Lombardi. Morphological Classification of Galaxies by Artificial Neural Networks. MNRAS, 259:8P, Nov 1992. doi: 10.1093/mnras/259.1.8P.

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2014.

J. Tinker, A. V. Kravtsov, A. Klypin, K. Abazajian, M. Warren, G. Yepes, S. Gottlöber, and D. E. Holz. Toward a Halo Mass Function for Precision Cosmology: The Limits of Universality. ApJ, 688:709-728, Dec. 2008. doi: 10.1086/591439.

M. A. Troxel, N. MacCrann, J. Zuntz, T. F. Eifler, E. Krause, S. Dodelson, D. Gruen, J. Blazek, O. Friedrich, S. Samuroff, J. Prat, L. F. Secco, C. Davis, A. Ferté, J. DeRose, A. Alarcon, A. Amara, E. Baxter, M. R. Becker, G. M. Bernstein, S. L. Bridle, R. Cawthon, C. Chang, A. Choi, J. De Vicente, A. Drlica-Wagner, J. Elvin-Poole, J. Frieman, M. Gatti, W. G. Hartley, K. Honscheid, B. Hoyle, E. M. Huff, D. Huterer, B. Jain, M. Jarvis, T. Kacprzak, D. Kirk, N. Kokron, C. Krawiec, O. Lahav, A. R. Liddle, J. Peacock, M. M. Rau, A. Refregier, R. P. Rollins, E. Rozo, E. S. Rykoff, C. Sánchez, I. Sevilla-Noarbe, E. Sheldon, A. Stebbins, T. N. Varga, P. Vielzeuf, M. Wang, R. H. Wechsler, B. Yanny, T. M. C. Abbott, F. B. Abdalla, S. Allam, J. Annis, K. Bechtol, A. Benoit-Lévy, E. Bertin, D. Brooks, E. Buckley-Geer, D. L. Burke, A. Carnero Rosell, M. Carrasco Kind, J. Carretero, F. J. Castander, M. Crocce, C. E. Cunha, C. B. D'Andrea, L. N. da Costa, D. L. DePoy, S. Desai, H. T. Diehl, J. P. Dietrich, P. Doel, E. Fernandez, B. Flaugher, P. Fosalba, J. García-Bellido, E. Gaztanaga, D. W. Gerdes, T. Giannantonio, D. A. Goldstein, R. A. Gruendl, J. Gschwend, G. Gutierrez, D. J. James, T. Jeltema, M. W. G. Johnson, M. D. Johnson, S. Kent, K. Kuehn, S. Kuhlmann, N. Kuropatkin, T. S. Li, M. Lima, H. Lin, M. A. G. Maia, M. March, J. L. Marshall, P. Martini, P. Melchior, F. Menanteau, R. Miquel, J. J. Mohr, E. Neilsen, R. C. Nichol, B. Nord, D. Petravick, A. A. Plazas, A. K. Romer, A. Roodman, M. Sako, E. Sanchez, V. Scarpine, R. Schindler, M. Schubnell, M. Smith, R. C. Smith, M. Soares-Santos, F. Sobreira, E. Suchyta, M. E. C. Swanson, G. Tarle, D. Thomas, D. L. Tucker, V. Vikram, A. R. Walker, J. Weller, Y. Zhang, and DES Collaboration. Dark Energy Survey Year 1 results: Cosmological constraints from cosmic shear. Phys. Rev. D, 98(4):043528, Aug 2018. doi: 10.1103/PhysRevD.98.043528.

L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008. URL http://www.jmlr.org/papers/v9/vandermaaten08a.html.

T. von Hippel, L. J. Storrie-Lombardi, M. C. Storrie-Lombardi, and M. J. Irwin. Automated Classification of Stellar Spectra - Part One - Initial Results with Artificial Neural Networks. MNRAS, 269:97, July 1994. doi: 10.1093/mnras/269.1.97.

Y. Wadadekar. Estimating photometric redshifts using support vector machines. *Publications of the Astronomical Society of the Pacific*, 117(827):79–85, jan 2005. doi: 10.1086/427710. URL https://doi.org/10.1086%2F427710.

M. S. Warren, K. Abazajian, D. E. Holz, and L. Teodoro. Precision Determination of the Mass Function of Dark Matter Halos. ApJ, 646(2):881–885, Aug 2006. doi: 10.1086/504962.

R. H. Wechsler, J. S. Bullock, J. R. Primack, A. V. Kravtsov, and A. Dekel. Concentrations of Dark Halos from Their Assembly Histories. ApJ, 568:52–70, Mar. 2002. doi: 10.1086/338765.

S. Weinberg. A new light boson? *Phys. Rev. Lett.*, 40:223–226, Jan 1978. doi: 10.1103/PhysRevLett.40.223. URL https://link.aps.org/doi/10.1103/PhysRevLett.40.223.

F. Wilczek. Problem of Strong P and T Invariance in the Presence of Instantons. *Phys. Rev. Lett.*, 40:279–282, 1978. doi: 10.1103/PhysRevLett.40.279.

I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

T. Ye, X. Wang, J. Davidson, and A. Gupta. Interpretable intuitive physics model. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 87–102, 2018.

M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 818–833, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10590-1.

Y. B. Zel'dovich. Gravitational instability: An approximate theory for large density perturbations. A&A, 5:84–89, Mar. 1970.

Q.-s. Zhang and S.-c. Zhu. Visual interpretability for deep learning: a survey. *Frontiers of Information Technology & Electronic Engineering*, 19(1):27–39, Jan 2018. ISSN 2095-9230. doi: 10.1631/FITEE.1700808. URL https://doi.org/10.1631/FITEE.1700808.

X. Zhang, Y. Wang, W. Zhang, Y. Sun, S. He, G. Contardo, F. Villaescusa-Navarro, and S. Ho. From Dark Matter to Galaxies with Convolutional Networks. *arXiv e-prints*, art. arXiv:1902.05965, Feb 2019.

B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2929, 2015.

F. Zwicky. Die Rotverschiebung von extragalaktischen Nebeln. *Helvetica Physica Acta*, 6:110–127, 1933.